University of Zurich[UZH]

Department of Informatics

Measuring and Analyzing Video Downloads in BitTorrent

Dissertation submitted to
the Faculty of Business, Economics and Informatics
of the University of Zurich

to obtain the degree of
Doktor / Doktorin der Wissenschaften, Dr. sc.
(corresponds to Doctor of Science, PhD)

presented by
Andri Filip Lareida
from Aarau, Switzerland

approved in September, 2017

at the request of
Prof. Dr. Burkhard Stiller
Prof. Dr. Tobias Hoßfeld

University of Zurich[UZH]

The Faculty of Business, Economics and Informatics of the University of Zurich hereby authorizes the printing of this dissertation, without indicating an opinion of the views expressed in the work.

Zurich, September 20, 2017

Chairwoman of the Doctoral Board: Prof. Dr. Elaine M. Huang

Acknowledgments

I WOULD LIKE TO EXTEND MY SINCERE GRATITUDE to a number of people, who supported me in the process of completing the thesis at hand.

First, I would like to thank my doctoral adviser, Prof. Dr. Burkhard Stiller, for dragging me into this adventure, for supporting me with advice and action whenever it was needed, and for keeping his humor in times when I lost mine. Also, I would like to thank my co-adviser Prof. Dr. Tobias Hoßfeld, for the valuable input and feedback to my research as much as the though but fruitful work package 7 meetings. Thank you.

I would like to thank all those who co-authored scientific work with me and who's contributions helped me improve and focus my research. Particularly, Dr. Thomas Bocek, for being my technical consultant and Dr. Martin Waldburger, for pushing me towards my first publication. As well as Valentin Burger, Michael Seufert, and George Petropoulos for the great team work in the SmartenIT project and all students I had the honor to supervise in their thesis or assignment work, for doing great work and contributing to my research. Thank you.

A very big thank you goes to Alexander Kübler who sacrificed his vacation time to proof read this thesis and give it the final edge.

I would like to thank all members of the CSG whom I had the pleasure to work with. Especially, to Dr. Christos Tsiaras, for being a great colleague and a true friend, to Dr. Fabio Victora Hecht, for introducing me to the art of server administration and picanha, Dr. Guilherme Sperb Machado, for our many boisterous discussions, and Patrick Poullie for sharing the pain and the laughs. Thanks a bunch.

My sincerest gratitude goes to my parents, Daniel and Ines Lareida, for the endless support and love you gave me and to my friend, Anita Bissig, for cheering me up in the rougher times and always believing in me.

Last, but not least, thank you Prof. Dr. Peter Athanas for your interest in my progress and the pep talks on the home stretch.

Abstract

BitTorrent (BT) is still widely used, with the most popular content shared being copyrighted. For the assessment of counter-measures, detailed and complete BT measurements are required. Therefore, this thesis investigates how to collect and analyze complete BT data. Through the comparison of prior BT measurement studies and their resulting data sets, a gap in existing measurements was identified. Specifically, measurements, or data sets, providing multiple samples per hour over weeks while still covering tens of thousands of swarms were missing. Therefore, this thesis makes the following contributions: (a) the design and implementation of a distributed scalable BT measurement system, termed VIOLA, which can collect data sets that fill the identified gap; (b) a data set covering 3 months, 70,000 swarms, and 3 samples per hour, proving that the VIOLA system works as designed and providing the basis for novel insights in BT behavior; (c) a method to transform the measurement data into graphs with different semantics building the basis for quantifying user- and content-centric properties of the BT system.

First, swarm size estimators, *i.e.,* simple and maximum likelihood estimators, are investigated through simulations and real world BT time series. Second, the collection of swarms through random draws is modeled accurately to orchestrate collection across multiple collectors. This problem is termed the BitTorrent Peer Collector (BTPC). To solve this problem, an analytical model to predict the number of tracker requests needed to collect a swarm is derived. Based on these insights the design of a distributed measurement system, termed Video Consumption in Overlay Networks (VIOLA), is presented and explained. Different storage options for the resulting data are discussed and tested.

To assess the capabilities of the VIOLA implementation and to collect a data set that goes beyond the state of the art, two measurement runs were executed in April 2015 and from May to July 2016. The 2015 measurement was executed with a simple scheduler not considering swarm size for collection, resulting in a data set covering 14 days and 5,000 swarms. With an adaptive scheduler system, three months of measurements were collected in 2016. Resulting in 98% of swarm collections (3 times an hour) collecting more than 95% of the swarm size reported by trackers. Compared to prior BT measurements, the VIOLA data set provides a new quality by combining the time, content, and user dimensions in an unprecedented way. A first analysis of the collected data shows that North America has fallen behind Asia and Europe in the total number of active peers compared to prior studies. The popularity of content, measured by maximum swarm size, shows an almost exponential distribution. Generally, tracker reported swarm sizes are lower than those collected from the Distributed Hash Table (DHT). The confiscation of the "Kickass Torrents" portal's

servers by the FBI, shows that a single measurement snapshot can be very biased and continuous measurements of BT swarms and sufficient level of detail, as provided by VIOLA, are necessary to draw valid conclusions.

To analyze the collected data and quantify the importance of individual countries, Autonomous Systems (ASes), and content a method to transform the tracker responses into a one-mode graph, allowing quantification of node centrality, is introduced. Two options for projecting the two-mode (bipartite) BT network graph, consisting of torrents and peers, are investigated. The first option is replacing the torrent nodes with edges between the peers, which were aggregated to countries and ASes with edges weighted according to the potential traffic exchanged with other ASes. In the country graph, the United States is the most central country in all metrics followed by GB and CA. Despite efforts to eradicate content piracy, the United States and Canada are important in the BT ecosystem. However, the AS network weighted by the potential traffic shows that the Philippines are responsible for most traffic during the whole measurement period. The second option is to replace peers with edges between torrents, resulting in a content centric network. This graph can be used to discover or recommend new content to users by following the edges of content they already consumed. The content network allows to rank content according to its centrality in the BT system through node centrality measures, providing a new angle on the popularity of content. The application of those measures to the VIOLA data set reveals strong weekly patterns, caused by the release of new episodes of the "Game of Thrones" show. Additionally, the release of a new episode caused a peak in the centrality of older episodes and also popular movies such as "Captain America Civil War". Thus, weekly patterns are a result of releases rather than a result of the day of the week.

The network analysis shows how the noisy and redundant data from the VIOLA measurement system can be transformed into a graph structure which can then be used to calculate measures. Besides the benefit of providing information on the relations between countries or ASes and between content, Social Network Analysis (SNA) measures are a way of monitoring a system over an extended period of time. Providing the basis to evaluate changes in the BT system, *e.g.,* due to anti-piracy measures.

Kurzfassung

BitTorrent (BT) erfreut sich immer noch großer Beliebtheit und wird vor allem zum Teilen von urheberrechtlich geschützten Inhalten verwendet. Um Gegenmaßnahmen und Trends zu beurteilen sind detaillierte und komplette BT Messungen notwendig. Folglich beschäftigt sich die vorliegende Dissertation mit der Frage, wie man solche komplette BT Daten erheben und analysieren kann. Durch den Vergleich früherer BT Messungen und der resultierenden Datensätze wird eine Lücke in bestehenden Messstudien aufgezeigt. Spezifisch fehlen Datensätze oder Messungen, welche mehrere Proben pro Stunde über mehrere Wochen bieten und immer noch zehntausende Schwärme abdecken. Aus diesem Grund leistet diese Dissertation folgende Beiträge: (a) den Entwurf und die Implementierung eines verteilten und skalierbaren BT Messsystems, genannt Video Consumption in Overlay Networks (VIOLA), welches fähig ist besagte Lücke zu schließen; (b) einen neuen Datensatz über 3 Monate, 70'000 Schwärme und 3 Proben pro Stunde. Dieser Datensatz beweist, dass VIOLA wie vorgesehen arbeitet und somit die Basis für neue Einsichten in das Verhalten des BT Systems und seinen Nutzern bildet; (c) eine Methode zur Transformation der Messdaten in verschiedene Graphen mit unterschiedlichen Bedeutungen and denen sich Inhalt und Nutzer bezogene Eigenschaften von BT untersuchen lassen.

Zuerst wird in Simulationen und mithilfe einfacher BT Messdaten untersucht, wie die Schwarmgröße mit simplen und größter Wahrscheinlichkeit basierter Verfahren geschätzt werden kann. Danach wird die Sammlung kompletter BT Schwärme durch zufälliges Ziehen mit Zurücklegen modelliert um die verteilte Sammlung zu koordinieren. Dieses Problem wird als BitTorrent Peer Collector (BTPC) bezeichnet, in Anlehnung an das Sammelbilderproblem. Zur Lösung dieses Problems wird ein analytisches Modell zur Vorhersage der Anzahl benötigter Anfragen (Ziehungen) hergeleitet. Basierend auf diesen Erkenntnissen wird das verteilte VIOLA System entworfen und implementiert. Verschiedene Persistenz Varianten werden diskutiert und getestet.

Um die Fähigkeiten des VIOLA Systems zu beurteilen und einen Datensatz zu erheben, der über die existierenden hinausgeht, wurden zwei Messreihen im April 2015 und von Mai bis Juli 2016 durchgeführt. Die Messung 2015 wurde mit einem einfachen Steuerprogramm ausgeführt, welches nicht zwischen verschiedenen Schwarmgrößen unterscheidet. Das resultierte in einem Datensatz, der 14 Tage und 5'000 Schwärme abdeckt. Mit einem adaptiven Steuerprogramm wurde eine Messung über 3 Monate in 2016 durchgeführt. Das Resultat zeigt, dass von allen Proben 98% der Schwärme zu mindestens 95%, der angegebenen Schwarmgröße gesammelt wurden. Verglichen mit früheren Datensätzen bietet VIOLA vor allem

bezüglich der zeitlichen Komponente neue Qualitäten. Eine erste Analyse der Daten zeigt, dass Nordamerika im Vergleich zu älteren Studien in der Anzahl an Nutzern hinter Asien und Europa zurückgefallen ist. Weiter zeigt die Popularität von Inhalten gemessen an der maximalen Schwarmgröße eine fast exponentielle Verteilung und deckt sich somit mit älteren Resultaten. Generell sind die Schwarmgrößen, welche von Trackern angegeben werden, kleiner als die Anzahl Adressen in der verteilten Hash-Tabelle (Distributed Hash Table (DHT)). Die Beschlagnahmung der Server des "Kickass Torrents" Portals durch das FBI zeigt, dass einzelne Momentaufnahmen nicht genügen um generelle Aussagen über BT zu machen, und dass kontinuierliche Messungen notwendig sind um valide Schlussfolgerungen zu ziehen.

Um die gesammelten Daten zu analysieren und die Wichtigkeit einzelner Länder, Autonomer Systeme (Autonomous System (AS)) und Inhalte zu quantifizieren, wird eine Methode vorgestellt um die Messdaten in einen einfachen Graphen zu transformieren auf welchem dann die Zentralität der Knoten berechnet werden kann. Zur Projektion des BT Graphen, bestehend aus Nutzern und Inhalten, gibt es zwei Möglichkeiten. Die erste Option ist, die Inhalt-Knoten mit Kanten zwischen den Nutzer-Knoten zu ersetzen, welche dann zu Ländern oder ASen mit Dateigrößen gewichteten Kanten aggregiert werden können. Im Länder-Graph sind die Vereinigten Staaten von Amerika am zentralsten, gefolgt von Großbrittanien und Kanada. Trotz erheblicher Antipirateriemaßnahmen spielen die nordamerikanischen Länder eine zentrale Rolle im BT System. Demgegenüber steht die prominente Rolle zweier philippinischer ASe, was bedeutet, dass der meiste Internetverkehr von diesen verursacht wird. Die zweite Option, Graphen zu erstellen, ist die Nutzer durch Kanten zwischen Inhalten zu ersetzen. Dieser inhaltsspezifische Graph lässt sich verwenden um Vorschläge zu berechnen, welche Nutzer interessieren könnten, basierend auf deren Historie. Weiter, kann man damit die Inhalte nach ihrer Zentralität ordnen und erhält eine neue Perspektive zur Popularität von BT Inhalten. Die Anwendung verschiedener Metriken auf diesen Graphen hat gezeigt, dass es starke wöchentliche Muster in der Beliebtheit von Inhalten gibt. Diese Muster erklären sich durch die Veröffentlichung neuer Episoden beliebter TV-Serien wie "Game of Thrones". Interessanterweise wirken sich neue Veröffentlichungen auch positiv auf ältere Episoden aus und auch beliebte Filme, wie "Captain America Civil War", sind davon betroffen. Daher kann man schlussfolgern, dass diese Muster eher mit der Veröffentlichung von Episoden als mit dem Wochenrhythmus der Nutzer zu tun hat.

Die präsentierten Netzwerkanalysen zeigen, wie redundante und rauschende Daten aus dem VIOLA System in einen Graphen transformiert werden können, auf welchem dann verschiedene Metriken zur Bestimmung der wichtigen Knoten berechnet werden. Diese Metriken bieten eine einfache und abstrakte Möglichkeit, das BT System über lange Zeiträume zu überwachen und somit die Effekte verschiedener Antipirateriemaßnahmen sichtbar zu machen.

Contents

Listing of figures

Listing of tables

I think there will be 20 years of evolution from linear broadcast to internet television.

Reed Hastings

1 Introduction

Continuous advances in broadband technologies enabled an increase of Internet connection speeds and increased number of subscriptions over the last decade in fixed and mobile access [25]. This advance has led to the emergence of on-demand video streaming services, such as Netflix, HULU, and Amazon Video, challenging traditional broadcast Television (TV) as well as digital media sales. Additionally, Peer-to-Peer (P2P) file sharing systems, of which BitTorrent (BT) is by far the most popular [54, 55], showed continuous increase of absolute network traffic in North America and Europe. The emergence of those Video on Demand (VOD) services significantly reduced the relevance of BT traffic. However, in Asia, BT is still causing 29.76% of aggregated (upstream and downstream) traffic, and in Europe's and Latin America's fixed networks it is ranked third of traffic producing applications [54]. According to [10] P2P file sharing accounted for 4,798 PB of Internet traffic per month in 2015. Besides the traffic aspect, file sharing in general and BT specifically is used to distribute copyrighted content which is either not available in a certain region or too expensive for some consumers. This content-piracy is harming movie production [18] but also the music industry is affected. However, concrete facts concerning BT usage are not publicly available since Internet Service Providers

(ISPs) do not disclose the numbers of copyright infringement incidents of their customers. Thus, BT is still a relevant and important research topic.

1.1 Revolution in Video Consumption Behavior

The home entertainment market is undergoing changes in the ways video content is consumed. Physical media sales, such as Digital Versatile Discs (DVDs) and Blu-Ray Discs (BDs), are losing importance as sales numbers decline while streaming revenue is rising [9]. This revolution is reflected in Internet traffic statistics [54], confirming the importance of on-demand entertainment for ISPs and content providers. VOD providers are not only competing against physical media and among each other, but also against file sharing systems like BT [27, 52]. File sharing platforms have superior content catalogs to VOD services with movies appearing on BT portals before their theatrical releases, *e.g.,* “The Revenant” [3]. For content to become available on VOD platforms more time is required [20]. Therefore, file sharing platforms are a good source for VOD to identifying the market potential of new content. The easiest way to monitor the popularity of BT content is to monitor the portals, typically providing an overview of the number of peers sharing a file.

Over the last years, file sharing peak traffic share has declined mainly due to the rapid growth of real-time entertainment, *e.g.,* VOD, traffic[54, 55]. However, in absolute numbers, file sharing traffic is still expected to increase over the next five years [10].

1.2 Content and Network Traffic

The Internet is comprised of Autonomous Systems (ASes) belonging to ISPs. ISPs typically own one or more ASes which collectively constitute the domain of an ISP. ISPs operate on different levels; the most simple distinction is access and backbone (Tier 1) providers. Access providers sell their services to end-users, *e.g.,* in the form of DSL or Cable subscriptions. Backbone providers sell transit connections to access ISPs, including transatlantic links that are unpractical for access providers to operate by themselves. Access providers seek to reduce their transit traffic cost by reaching a peering agreement with other access providers [46]. Peering agreements typically do not involve payment but an exchange of traffic between the partners’ domains. Large content providers seek peering agreements with access providers

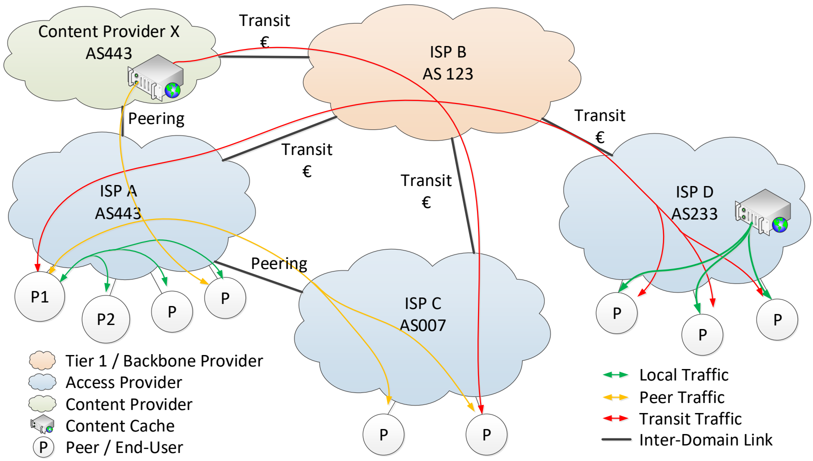


Figure 1.1: Autonomous Systems and the potential file sharing traffic flow between them.

directly [11]. However, access providers, operating their own content delivery platform [11, 71], have no incentives to peer with content providers. This situation leads to conflicts between ISPs and content providers. Figure 1.1 provides an overview of ASes and their different types of connections and the possible placement of content caches.The SmartenIT project [73] investigated the optimization potential of overlay applications and the potential of overlay applications to optimize data center to client traffic.

VOD is responsible for the largest portion of downstream traffic [10, 54, 55]. To achieve this, content providers serve their content from different locations. There are three options to deliver traffic to end-users:

1. Deliver from the content provider's data center through a transit network. This option is indicated in Figure 1.1 by a red arrow AS443 delivering data through AS123 to an end-user in AS007, meaning that this is, trafficwise, the worst option.

2. Deliver content from the content provider's cache through a peering link to the end-user in a neighboring AS. This option is indicated in Figure 1.1 by the yellow arrow leaving AS443 to AS007, which is a good option as the peering is typically free [11, 46].

3. Deliver content from a content provider owned cache located on the premises of an ISP [45]. This option is indicated in Figure 1.1 by the green arrows in AS233 as the content is locally served and no inter-domain traffic is caused.

The availability of options 2 and 3 confirm that the technology and processes to handle VOD traffic and reduce its impact on ISPs' operative expenses exist. However, as numerous examples [22, 58, 71] prove, the political aspect of deploying a technical solution remains a challenge. More progressive approaches [34, 35] suggest caching content at the edge of the network within user controlled devices, such as their personal computers or home routers.

P2P traffic has a different character than VOD traffic since it originates at the access level of the Internet. Members of P2P systems are simultaneously downloading from and uploading to their peers. Thus, the impact on inter-domain traffic is potentially larger than with VOD traffic. However, the delivery options are analogous to the VOD case. Ideally, peers from the same AS trade pieces of a file and no inter-domain traffic is caused. This case is illustrated in Figure 1.1 by the green arrows between peers in AS443. The less optimal option is that peers from peering ASes exchange data, which does not induce cost, as the peering is free, but the capacity of the inter-domain link needs to be high enough to handle the traffic. Expanding the capacity of a peering link often involves re-negotiations of peering agreements and the cost of additional hardware and infrastructure [11, 46]. This option is illustrated in Figure 1.1 by the yellow arrows connecting peers from AS443 and AS007. The last and least optimal case is the traffic flowing through an expensive transit link. This is illustrated in Figure 1.1 by the red arrows crossing AS123 to connect peers from AS443 and AS233. Studies [17] show that BT is not locality aware, *i.e.,* BT does not account for network topology when making connections.

1.3 VOD and Content Piracy

VOD providers claim to be the answer to the piracy problem and P2P file sharing, arguing that piracy is reduced where big players start offering their services. According to Netflix's CEO Reed Hastings, piracy prepared users for VOD services as they learned what to expect from a streaming service [2]. By offering a better experience than file sharing systems, VOD providers convince users to move from file sharing to using and paying for their services. However, the reduction of piracy is relative since the traffic data shows only traffic shares [54], saying nothing about absolute numbers of file sharing traffic. However, the relative increase of VOD traffic does not imply that piracy is decreasing. Furthermore, those traffic shares were measured during peak times, which does not necessarily reflect total traffic shares. Traffic shares of single applications show that BT is still the most traffic generating application in Asia's fixed networks and third most in Europe's and Latin-America's fixed networks [54, 55]. Thus, it can be stated that file sharing piracy remains an issue in most parts of the world.

When entering a new territory VOD services face competition from the existing television providers as well as from piracy [52], mainly in the form of BT. The time span for content to become available on VOD platforms is typically longer than the time it takes the same content to be available on file sharing platforms[3, 20]. Therefore, it is important for VOD providers, to expand their catalogs in the new regions to convince more file sharers of the advantages of VOD platforms. File sharing portals can be used to identify popular content, if the portal is regionally bounded, *e.g.,* a portal exclusively in the Dutch language, it can indicate regional content preferences. However, portals alone do not offer detailed insights in what content is downloaded in which regions. Thus a more detailed view on pirated content downloads can help improve VOD catalogs and fight privacy.

1.4 Reasearch Questions and Contributions

The current situation as presented herein is not satisfying. Content piracy is still a relevant problem which is not yet solved and accurate methods for measuring it are not available. First, the traffic generated by BT is still high, and it is not optimized to avoid expensive inter-domain links like VOD traffic, which is typically cached locally. Furthermore, the use of dedicated servers in foreign countries to avoid local legislature increases inter-domain traffic even more, and this additional traffic is not accounted for in file sharing statistics. Second, BT, being the most used file sharing application, cannot be simply shut down, and continues to spread content before it is available through legal channels. VOD providers are expanding into new territories, requiring negotiations with content owners, *e.g.,* movie studios. Thus, providing content that attracts the most users is critical to success in a region. File sharing portals are already used as an indicator of popularity for content. However, this approach is very limited as the level of detail in download statistics from those portals is not high enough to identify detailed regional differences. Therefore, this thesis investigates the following research questions to provide the data and tools to monitor the state of the content piracy problem.

1. What is state of the art in BT measurement and monitoring?
   (a) What are the most critical measurements and what are their conclusions?
   (b) Where is the gap in existing BT measurement methodology?
2. How can this gap in BT measurements be closed?
   (a) How can all peers sharing a file be identified?
   (b) How can a BT measurement system be built?
3. Can BT be measured accurately?
   (a) What accuracy can the designed system achieve?
   (b) What resources are required to achieve a certain accuracy?
   (c) Does the collected data confirm prior findings?
4. What insights can be gained with such a dataset?
   (a) How can the raw measurement data be transformed to apply standard methods to it?
   (b) Which countries are important in BT
   (c) Which ASes are creating the most traffic?
   (d) Which content has the largest influence on the users?

This thesis' structure is motivated by those questions. Thus, the contributions provided by this thesis can be summarized:

- A survey of existing BT measurement and monitoring studies. Providing an overview of the key aspects of the most important BT measurements and showing the gap in existing BT measurement methodology.

- To close this gap, the collection of BT peers, *i.e.,* the BitTorrent Peer Collector (BTPC), is analytically analyzed and verified in experiments.

- The Video Consumption in Overlay Networks (VIOLA) measurement system applying those findings is developed, and its key components' design and implementation choices are detailed.

- A dataset collected over three months, comprised of $\approx 70,000$ movies and TV shows and more than $110,000,000$ unique Internet Protocol (IP) addresses.

- A method, capable of abstracting this amount of data to a graph of users or contents, is presented. Social Network Analysis (SNA) methods are applied on those graphs to identify important ASes, countries, and contents.

Those contributions improve the understanding of BT measurements, provide a solution to apply this understanding in practice, and give insight into the changes of BT over time. The collected data can be used to monitor the effects of anti-piracy measures taken in certain countries, such as the shift of traffic to countries or ASes under more liberal anti-piracy legislature. Furthermore, it allows ISPs to better understand the traffic produced by BT users and to take appropriate measures,*e.g.,* caching or expanding link capacity, before the release of popular TV show or movie.

1.5 Thesis Outline

The remainder of this thesis is structured as follows. Chapter 2 discusses the relevant background information on the BT protocol, its ecosystem, and on challenges of measuring P2P systems in general. Furthermore, Chapter 2 provides a survey of existing BT measurements and highlights the gap existing in current measurement methodology, which is closed in this thesis. Chapter 3 provides an analytical investigation of the BTPC problem and verification based on real world measurements. The second part of Chapter 3 presents the key design and implementation aspects of the VIOLA measurement system. Chapter 4 presents measurement results obtained by applying the VIOLA system. Two data sets were collected, two weeks in April 2015 and three months in summer 2016, of which the metadata is described. Furthermore, the quality of the data in those two data sets is analyzed and compared to each other and previous measurements. Chapter 4 contains all required information for re-using those data sets. In Chapter 5 the 2016 dataset is abstracted to a one-mode graph on which SNA measures are applied to identify popular and important content connecting different clusters of potential VOD customers. Finally, the thesis is concluded in Chapter 6.

If I have seen further it is by standing on the shoulders of Giants.

Isaac Newton

2 Related Work

THE BITTORRENT (BT) NETWORK has been used for file sharing for more than a decade and sparked research interest since the publication of the original paper [13]. BT research focused not solely on the protocol and incentives, but also on measuring the live system. Therefore, the necessary background information on BT and its ecosystem are presented, followed by the background on BT measurement challenges and measurement categorization. Since several measurement studies were conducted covering the dimensions: time, content, and users at different levels of detail. A survey of the most prominent examples of those previous measurements is provided. One notable example for such a purpose is locality awareness, aiming at connecting peers that are physically close, to reduce inter-domain traffic. Various locality approaches tailored to BT are summarized, providing an overview of the state of the art. Finally, previous BT measurements are visualized based on their level of detail in those three dimensions. It is concluded that currently available measurements do not offer all levels of detail needed to investigate content distribution on a global scale.

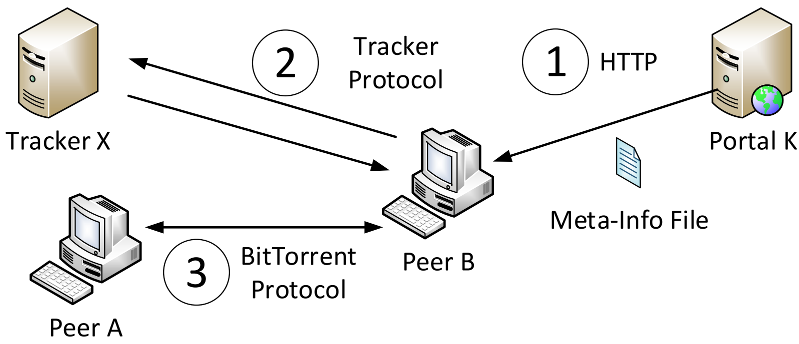


Figure 2.1: The entities of the BT ecosystem and their relations.

2.1 The BitTorrent Ecosystem

BT is a file distribution system built on Peer-to-Peer (P2P) technology featuring properties like scalability, reliability and flexibility [59]. The basic idea behind P2P is that every client is also a server and therefore all actors are called peers because they are equal. BT implements this paradigm by applying a "tit-for-tat" principle [13], which encourages peers to upload to improve their download speed. Therefore, the more peers download a file, the more upload bandwidth will become available neutralizing any potential bottlenecks in servers or data centers. The BT ecosystem's main components are introduced in this section as BT is the most popular P2P file sharing system.

Figure 2.1 depicts the three steps comprising the traditional BT ecosystem. First, a meta-info-file needs to be downloaded from a BT portal, which is a website. Second, a peer queries trackers found in that meta-info file to find other peers using the BT's tracker protocol. Third, peers establish a direct connection exchanging file parts using the BT protocol. The following terminology is used:

File: can be a movie, software, or any other electronically stored information.

Sharing: offering a file to other participants of the BT network.

Torrent: a file being shared in BT. It is identified by the info-hash (*cf.* Section 2.1.2).

Peer: an entity that takes part in BT. Typically, a personal computer running a BT client software.

Provider: a peer that has at least one piece of the torrent it is sharing.

Tracker: a server or service that keeps a list of peers for torrents and provides random samples out of those lists.

Node: an entity that participates in a Distributed Hash Table (DHT), typically using the same client software used for peers.

2.1.1 BitTorrent Portals

BT portals, or indices, are web-based portals which allow users to browse files currently being shared. Those files are typically categorized into categories, such as TV shows, movies, and music. Furthermore, a portal offers *.torrent* files for download or it provides magnet links which allow to download the required parts of a *.torrent* file from other peers. The two most popular BT portals [19] are "The Pirate Bay" [67] and "Kick Ass Torrents" [30]. Figure 2.2 illustrates how such a portal can look like, in this case, the screenshot is from "The Pirate Bay". Besides BT protocol specific information, they provide further details regarding the files they offer, such as title, date published, comments, popularity, and more.

However, portals are not part of the BT specification since the protocol does not specify how *.torrent* files are exchanged. Portals cannot be considered to be P2P systems due to their centralized (web server & database) nature. But BT portals are important for users to share files through BT, enabling the discovery of newly published torrents as well as popularity scores of torrents. This property is strength and weakness simultaneously. While the lack of a built-in search and discovery mechanism makes the BT system dependent on external resources it also decouples the file distribution from discovery making BT also suitable for private file sharing [5]. There exist several private BT portals and trackers which can be joined only on invitation.

2.1.2 .torrent or Meta-Info Files

Figure 2.3 shows the elements of a meta-info file, usually called *.torrent-file*. A *.torrent-file* is a B-encoded dictionary which maps a string key to either a string, a number, a

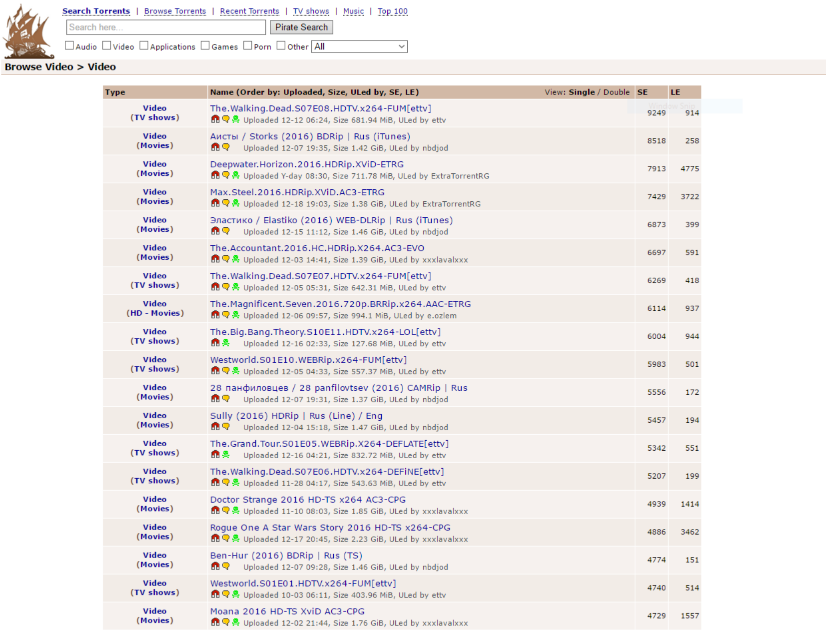


Figure 2.2: Screen shot of the Pirate Bay portal.

list, or a dictionary typed value. It contains all the necessary meta-information about a shared file, termed torrent. The files are typically downloaded from a BT portal and stored with the *.torrent* file extension. The important parts in a *.torrent* file are the *announce* or the *announces* and the *info* keys. The *announce* key maps to a single tracker Uniform Resource Locator (URL) string where the *announces* key maps to a whole list of tracker URLs, the *announces* key is optional but widely used. The *info* key maps to another dictionary that contains the details of the file(s) being shared. This is the part that can be downloaded from peers directly through the magnet-link extension.

BT splits files into pieces for efficient distribution. The info dictionary contains the number, length, and SHA-1 hash values of each piece. Under the piece length value, the length in bytes is stored, this is typically a multiple of 2. The pieces' value stores a concatenation of each pieces' 20 Byte SHA1 hash. Thus, it can be told how many pieces the file was split into, by dividing the number of bytes in the pieces' value by 20. Furthermore, the file size can be calculated from the piece length and the number of pieces. However, there can be inaccuracies resulting from padding

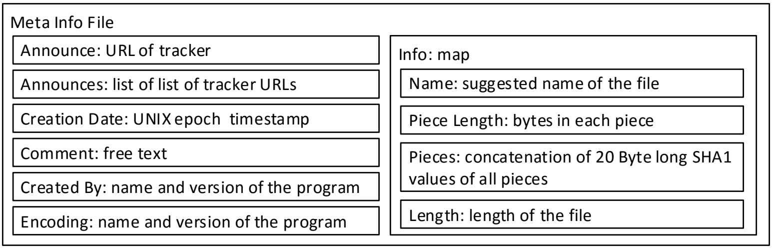


Figure 2.3: The main contents of a meta-info file for a single file download.

the last piece if the actual file size does not exactly fit. The info-hash is the SHA1 hash of the whole info dictionary, which is used as the identifier of the file, called info-hash. The info-hash is sent to trackers to identify the swarm from which peer addresses are requested.

2.1.3 Trackers

One important problem to solve in P2P systems is the bootstrapping process, or a peer joining a group of peers sharing a file, termed swarm. In BT, this problem is solved by the use of trackers which act as brokers between peers. To join a swarm, a peer must query at least one tracker to discover members of the swarm. There are two types of trackers: centralized and de-centralized. Centralized trackers are servers which store lists of peers sharing a certain file. De-centralized trackers are built on DHTs which belong to structured P2P systems [59].

2.1.3.1 Tracker Protocol

To discover providers, peers that are sharing a file, a peer needs to query a tracker. A tracker is either contacted through an HTTP GET message or a binary UDP based protocol. Either way, the URL of the tracker service can be found in the first part of the meta-info file. The most important request parameter is the info-hash which identifies the file. Further required parameters are: Peer-Identifier (ID), Internet Protocol (IP) address, and port number. However, trackers typically use the source address of the packets to register a peer since it might be behind a Network Address Translation (NAT) gateway. Additionally, the total bytes downloaded, the

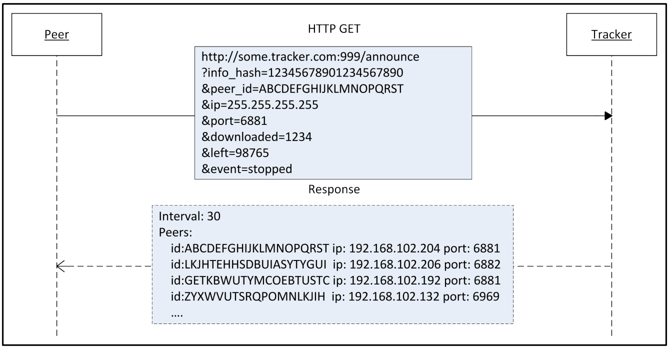


Figure 2.4: Example communication between a peer and a tracker.

bytes still left, and the type of the event are sent with a request. The event parameter gives the reason for the request which can be: started, stopped, completed, or blank, which means that it is a regular announcement done in the interval the tracker specifies in its response. There are additional parameters which are not required but are documented in [69].

Figure 2.4 gives a sample of the contents of a request and an answer. The answer is simply a list of peer addresses, consisting of an ID, an IP address, and port. Furthermore, the response includes the announce interval, telling peers how many seconds they should wait between two consecutive requests.

2.1.3.2 DHT-Tracker Extension

The BT DHT-Tracker extension is a fully distributed tracker which stores peer addresses under the info-hash. Two implementations exist: the Azureus Distributed Hash Table (AZDHT) [72] and the Mainline Distributed Hash Table (MDHT) [60]. Both versions are based on the Kademlia DHT [14], however, the latter is the official protocol extension of BT.

A DHT works like a normal hash table as it stores and retrieves values identified by a key. In its distributed flavor each participating node is responsible for a portion of the address space where a node is an instance of the DHT implementation. Each node has a 160-bit address which corresponds to the length of the info-hash. A node is responsible for the keys closest to its identifier. Thus, the more nodes are part of a DHT, the smaller is the address space for which an individual node is responsible. Closeness in Kademlia is defined by an Exclusive Or (XOR) metric [44], thus, the distance between two nodes equals the XOR of their IDs. Furthermore, keys are replicated to the 20 closest nodes to increase the robustness against failing nodes.

Table 2.1 compares the queries used by the two implementations used in BT and the Kademlia standard. Both AZDHT and MDHT stick close to the original query set. The queries may have different names in the different implementations, but the functionality is the same. Only the Azureus implementation adds a `KEY_BLOCK` query which can be used to request blocking and unblocking of keys. How this exactly works was not documented at the time of writing. The two BT implementations added an error message as a possible return which is not a query and is therefore not in the table.

Another difference is the bootstrapping process. The AZDHT uses a hard coded URL `dht.vuze.com:6881` to contact the bootstrapping node. To reduce the load on the bootstrapping node AZDHT contacts are saved upon shutdown of the client and peers discovered from other sources can also be used to bootstrap the AZDHT. The MDHT stores several known nodes in the *torrent-file* besides the static bootstrap node that can be reached under `router.bittorrent.com:6881`. Those nodes are typically either original seeders or special bootstrap nodes like the one used in the AZDHT. A few clients, like Vuze, support AZDHT as well as MDHT, the majority of clients support only MDHT.

Kademlia	Mainline	Azureus
PING	PING	PING
STORE	ANNOUNCE_PEER	STORE
FIND_NODE	FIND_NODE	FIND_NODE
FIND_VALUE	GET_PEERS	FIND_VALUE
N/A	N/A	KEY_BLOCK

Table 2.1: Comparison of the two BitTorrent DHT implementations and the Kademlia standard queries.

2.1.4 BitTorrent Protocol

After downloading a *.torrent-file* and receiving a list of peer addresses from a tracker, a peer will contact the peers in the list which are potential providers, peers that can provide pieces of the torrent. Peers which have not completed the download

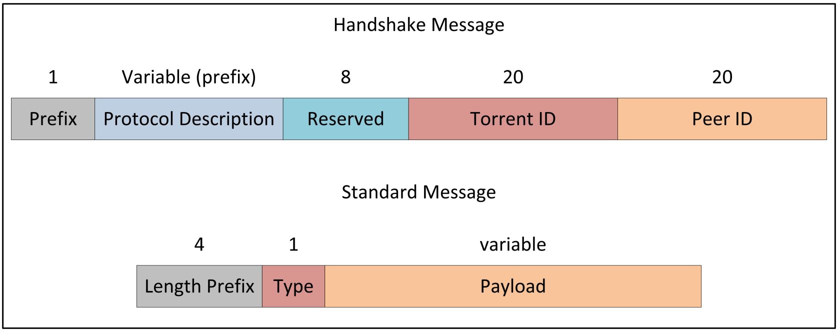


Figure 2.5: The two basic message types used in the BitTorrent protocol.

are called leechers, and those who are only uploading are called seeders. The BT protocol [12] governs the communication between peers.

Figure 2.5 shows a handshake message which peers exchange to connect to each other. A peer can contact multiple providers simultaneously. The handshake message for the standard BT protocol begins with a one-byte integer length prefix. The prefix indicates the length of the following protocol description which for the standard protocol is "BitTorrent protocol". Then a byte of zeros follows, this space is reserved for protocol extensions which are partially used already. The last two fields are the torrent ID and the Peer ID, both 20 bytes long. The torrent ID is the same info-hash as described before. The Peer ID is just an identification string; the BT protocol does not give any rules on how to determine it. The standard message can be used for any communication. It consists of a length prefix, indicating the total length of the message, a type field, indicating what the message is used for, and the payload, containing the actual message.

Figure 2.6 shows a sample of a communication flow between two peers. After receiving a handshake, the provider immediately returns a handshake with its torrent and peer ID. If either the peer or the provider notices that the torrent ID is wrong or it already has a connection to that peer related to that torrent, it drops the connection [17]. Wrong in this case means that either the peer does not share that particular torrent or the format is wrong. The provider's handshake reply is followed by the bit-field message indicating which pieces of the file it can offer for download. Each piece of the file is represented by a bit, which is set to 1 if the respective piece is available from that provider.

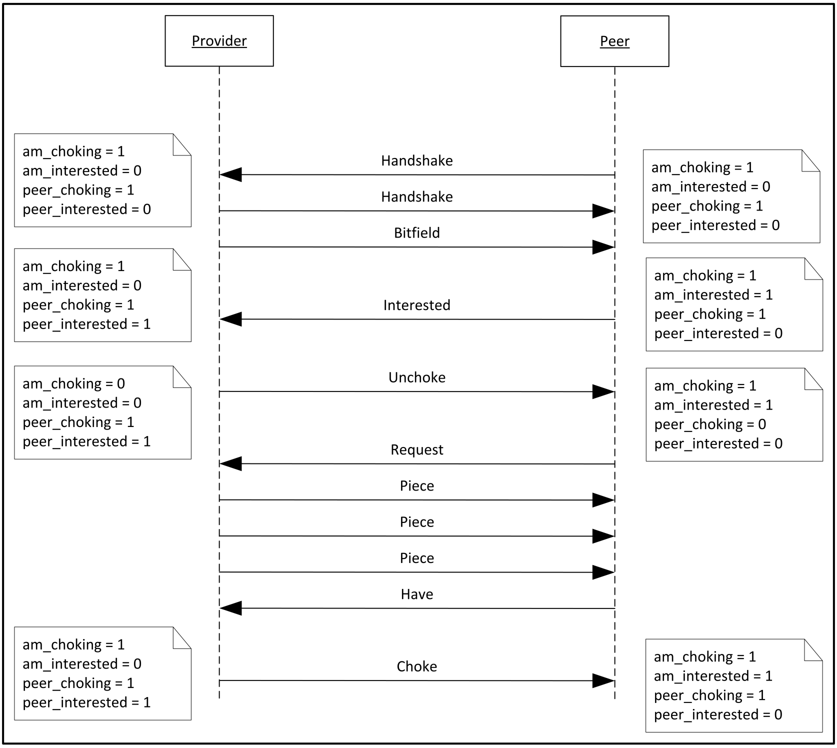


Figure 2.6: A typical message exchange between a peer and a provider following the BT protocol.

Figure 2.6 also shows the different states a connection can have on both ends, *e.g.,* a peer can choke a connection if it has no capacity left. Choking means that a peer serves no more pieces to its neighbor. If capacity becomes available, the provider sends an unchoke message to the client. A newly created connection starts as choked on both sides. According to the "tit-for-tat" mechanism, peers unchoke those peers, which have uploaded the most pieces recently. The interested flag tells if a neighbor wants to download a piece. There are also messages to set the interested flag or remove it. Therefore, a peer must maintain four state bits, one for its choke state, one for its interested state and two for the provider's states. The states of the sample connection between the peer and the provider are shown in the boxes.

Upon receiving a bit-field message, the peer determines if it is interested in a piece of the provider. It updates its link state to interested and it sends an interested message to the provider, which uses this message to update its link state. The provider knows now that as soon as it unchokes the peer, it will start sending requests. The provider answers the request by sending a portion of the requested piece since pieces are chopped into smaller blocks for easier transport. After the peer has received a complete piece, it sends a have message to all of its neighbors to let them know that it has that piece. Eventually, some neighbor may get interested in that piece after receiving a have message. Based on this new piece, the peer has to adapt its link states, maybe it is not interested in a peer anymore and therefore has to send a "not-interested" message. This process continues until a peer has completed the download. Then it will start seeding, that means the peer will serve pieces without downloading.

2.1.4.1 Peer Exchange

The idea behind Peer Exchange (PEX) [40] is to reduce the load on trackers and improve the connectedness of swarms. A peer sends a list of its neighbors to its neighbors. Thus, a peer can receive new peers not only from the tracker but also from other peers. PEX allows a swarm to keep together even if the tracker fails and it can also reduce the load on a tracker. However, PEX can not substitute a tracker completely since it does not have bootstrapping capabilities. To benefit from PEX a peer must be in a swarm already.

Experiments have shown that PEX can improve download speed [74] but also that PEX messages have a significant degree of redundancy. There are two main implementations of PEX: Azureus PEX (AZPEX) and uTorrent (UTPEX). The main difference is that UTPEX sends separate lists for IPv4 and IPv6 peers. Developers of both implementations have agreed to send a maximum amount of 50 peers per message [68]. They also agreed that messages should only be sent every minute. Research on PEX has shown that peer discovery is very quick for the first 30%-40% of the peers in a swarm but slows down and after 3000 seconds still has less than 60% of the total peers discovered [74]. Thus, with progressing peer discovery, it gets increasingly difficult to discover more peers of a swarm.

2.2 BitTorrent Measurement Background

Much work on BT measurements has been published, of which a selection is detailed in this section. Before going into detail on prior measurements general challenges of

P2P and BT measurements and their classification are presented. Finally, the largest or most important work in those categories is presented.

2.2.1 P2P Measurement Challenges

Overlay networks have received attention in research since their emergence, in the form of Napster and Gnutella, in the late 1990s [56]. Specifically for BT measurements, respective techniques, practices, and pitfalls are well known and documented [31]. Therefore, this section shall go into detail of general challenges in measuring P2P systems and cover challenges specific to BT.

2.2.1.1 Lack of Overview

The decentralization of distributed systems has three principal advantages: scalability, reliability, and flexibility. However, in a fully decentralized system, each entity has its subjective view of the system. Thus, a complete and objective view of such a system can only be gained by combining subjective views. In a DHT, for instance, it is not trivial to find the total number of nodes currently participating. The number can be estimated with the information present on one node [33]. However, the result might be biased.

2.2.1.2 Network Address Translation

P2P systems are mainly used by home users who typically get assigned a single dynamic IP address to the entire household. As a consequence, home routers translate the private IP addresses to the public address assigned by the Internet Service Provider (ISP) combined with a port number. NAT is an issue for P2P systems in general, as certain ports have to be forwarded by the router to a client which is listening for incoming connections. The Universal Plug and Play (UPnP) protocol solves the issue of dynamic port forwarding, but it is not always implemented or activated on home routers. In professional networks, *e.g.,* in universities or offices, UPnP is typically not supported. The same effect is created by firewalls blocking incoming ports which are also common in professional networks.

For P2P measurements this has implications. A certain amount of peers will not be reachable from the internet, limiting the possibilities of connecting to clients to find out more about the client software used or the connections a peer has with other peers. Studies found that up to 57.8% of peers in a BT swarm are not reachable due to being connected through a NAT device or a firewall [17].

2.2.1.3 IP addresses

Typically P2P users use the system from their homes through a Digital Subscriber Line (DSL) or cable connection. A property of those connections is that the assigned IP address changes once in a while. Therefore, identifying peers by their IP address can introduce inaccuracies in long-term analysis. According to a study [41] IP addresses are usually changed when a new session is opened, thus, especially dial-up connections suffer from frequent address changes. However, IP addresses are typically not changed during an active session allowing to identify a peer during its session which is usually longer than an internet browsing session.

2.2.1.4 Tracker and Coupon Collector

The BT network is orchestrated by trackers, which serve a maximum of 50 peers randomly selected from their pool of known peers. Therefore, the random peer selection results in redundancy of returned peers, especially for large swarms requiring many samples. Discovering all peers of a single swarm is an instantiation of the coupon collector problem [75] (*cf.* Chapter 3).

2.2.2 Categorization

BT measurement studies can be categorized in three different categories: macroscopic, microscopic, and complementary [31]. Macroscopic studies cover thousands of swarms but only in limited detail. Microscopic studies capture the detail of a few swarms. Complimentary approaches go beyond the basic measurement possibilities. Capturing the entire BT system in full detail is considered to be impossible due to the enormous number of peers and connections between them. BT measurements are always a tradeoff between resolution and coverage.

2.2.2.1 Macroscopic Measurements

Macroscopic studies of the BT network cover high-level facets of participating nodes (or users) for large parts of, or even the whole overlay network. However, those measurements do not provide any details on overlay nodes' connections or performance. Technically, these studies mostly use crawlers for BT portals (such as Kickass Torrents [30]) and trackers but do not connect to overlay nodes themselves to collect data. An example of a distributed BT measurement is [24] based on [23], which collected information by crawling trackers and index sites. However, snapshots or only high-level data were collected. Only one of their experiments recorded detailed IP addresses over 88 hours from 16 swarms. Another study [53] was measuring up to 100,000 swarms in snapshots and, thus, is limited in the time dimension.

2.2.2.2 Microscopic Measurements

Microscopic studies provide highly detailed insights into smaller parts of the overlay network, *e.g.*, inside a group or a swarm [17]. Unfortunately, those measurements do not document a complete picture of a network, mainly due to the effort that is required to collect those detailed insights for thousands of swarms. Microscopic studies typically follow a white box approach, in which the protocols under investigation need to be altered for specific interactions with the measurement infrastructure. Thus, in the case of overlay networks, this approach involves a reimplementation of parts of the overlay management protocols, *i.e.,* the BT protocol. These altered clients are used to connect to overlay nodes and to receive internal information about them, such as client type, version, or bit-fields, used to advertise what pieces of a file are available from a peer. With this technique, connections between overlay nodes in traffic flows can be deduced [17]. Like their macroscopic counterparts, microscopic studies typically look at snapshots of the network only. When connecting to overlay nodes, the problem of unreachable nodes arises due to NATs applied in many networks. Thus, even on a small scale, it is very hard or even impossible to achieve a complete picture of connections and their performance between peers. However, this is not considered to be problematic for VIOLA, since it will collect data on a lower level of detail and, therefore, does not require to be able to connect to nodes at all.

2.2.2.3 Complementary Approaches

Complementary approaches did augment those, micro- and macroscopic, measurements by adding DHT (Distributed Hash Tables) functionality to their crawlers [28] or by the use of plugins [8] for popular BT clients. [51] used vantage points to collect a detailed dataset on peers and their interconnections. However, the collected data does not allow to identify the content shared, which is a drawback as it does not allow conclusions on content popularity.

2.3 Survey of Previous BitTorrent Measurements

The most important prior BT measurements are discussed to provide an overview of the state of the art before this thesis. This survey focuses on macroscopic studies as these are best suited to provide large-scale data to investigate location local content popularity and piracy, *i.e.,* data containing the three dimensions: time, content, and users. In BT the time dimension includes the online time of peers and torrents but also the sampling rate and the total time covered by the measurement. The content dimension is reflected by torrents which are metadata of the content that is being downloaded. In P2P systems, users are typically identified by IP addresses and port numbers. The BT measurements presented differ in the level of detail they cover in each of those dimensions. To clarify these differences, those measurements are summarized and compared according to the parameters: measurement type, detail of data, samples taken, time covered, and the number of swarms measured. For completeness, one interesting example of a microscopic study is also presented. Several, of those studies, can be considered complementary according to the classification in [31]. However, complementary techniques can be used for macro- and microscopic measurements. Therefore, the measurements are classified in macro and micro.

2.3.1 On Blind Mice

The "Ono" plugin [8], further described in Section 2.4.2, was used to create vantage points in the BT network at the peers that installed it. Since "Ono" was installed in the client software, it was able to gather details of connections a peer was having, but for privacy reasons, the data does not allow to identify content shared. The data was collected mainly to understand the network impact of BT by analyzing the paths between peers, which is described in [51]. The "Ono" measurement can be classified as complementary and microscopic since it collects traffic and connection information by deploying a software component to BT users.

The data gathered for [51] was reportedly collected by more than 1,260,000 installations of the "Ono" plugin. However, the paper states: "Altogether, our dataset contains traces from more than 500,000 IPs located in 3,150 Autonomous Systems (ASes)" [51], which must mean that only 500,000 of the 1,260,00 installations were active during the collection. Data was collected between November 2008 through to November 2010, although not continuously. The study used traceroute data which was collected from randomly selected neighbors of vantage point peers. As the data is only described on a high-level it is not possible to say what exactly was collected.

The results of the paper [51] showed that: BT was evolving during their measurements (2008 - 2010), a significant amount of traffic stays local, and it is not feasible to measure P2P systems only in higher tier networks as more than 50% of the traffic is routed through smaller transit ISPs. The locality of traffic result contradicts some other work, and no explanation is given to that. The "Ono" plugin can be considered a microscopic study as it actively changed parts of the client software and it reveals also detailed peer interconnection information. However, the amount of data collected withing the "Ono" study leads to the perception that a macroscopic measurement had been performed.

2.3.2 BT AS Structure

[23, 24] present an extensive collection of BT data sets from the years 2008 and 2009. Three types of measurements were used to acquire those data sets: portal parsing, tracker monitoring, distributed monitoring. Portal parsing was done using a crawler that went through the portal and collected the seeder and leecher numbers for each torrent found. Tracker monitoring was done by issuing scrape requests to a tracker which return seeder and leecher numbers of the swarm. Distributed monitoring made use of global research facilities to issue many announce requests to trackers and collect the peers returned by those.

First, the swarm size results from all measurements are used to compare swarm sizes between the data sets. This comparison shows that different data sets show different swarm size distributions. The first discovery is that the dataset with torrents released in the last 24 hours had the biggest swarms, except the most popular dataset which has naturally the highest swarm sizes. The second important observation is that Movie and TV data sets have higher swarm sizes than the Music dataset. A reason for this could be the smaller file size which leads to shorter session times. The last observation was the swarm sizes following the Pareto principle, *i.e.,* 80% of the peers belong to 20% of the swarms, meaning that most swarms are very small.

Second, with the swarm size results from the portal monitored TV dataset, the time-dynamics of swarms were investigated. The results do not show a correlation between swarm size and swarm fluctuations. However, the swarm size evolution showed distinct diurnal patterns for some content which was mainly of regional interest due to its language. Auto-correlation analysis showed that only about 5.7% of the swarms show a distinct day-night behavior. Unfortunately, the authors do not tell what fraction of peers is in those 5.7% of swarms.

Third, the biggest part of the analysis revolves around the AS structure of swarms. For this purpose, only the distributed monitoring data sets could be used as IP addresses are required to determine the AS or country which they originate from. The first conclusion related to the AS structure was that bigger swarms have a higher potential for improving traffic locality, which leaves an optimization potential of 65% for the Movie dataset. The authors found that the AS size correlates to the number of peers in an AS only for large swarms. However, this result is likely biased as the distributed monitoring takes several hours and, thus, some ASes are experiencing a peak while others experience a through. Also in the case of a swarm with little fluctuation, it was found that most peers come from one AS hosting servers. The article explained this effect with fake peers being introduced to disturb the system. However, it is more likely that so-called "seed boxes" are hosted in that AS which are rented by users to have faster downloads or to run their torrent client 24/7.

This work presented a careful measurement and analysis on the BT network. Different macroscopic measurements are used to gain insight in the AS structure of swarms. The measurements using portal crawling and tracker monitoring cover a large number of swarms and up to 36 hours but reveal only high-level information about those swarms. Distributed monitoring gives detailed information on swarms allowing to investigate their AS and geographical structure. However, either only snapshots were taken (which take hours to collect) or a very limited amount of swarms was covered. Furthermore, the article finds that the distribution of Peers over ASes follows a power law distribution which is again observed in the popularity of swarms as 20% of the swarms contain 80% of all peers.

2.3.3 Unraveling the BitTorrent Ecosystem

[76] collected an overview of the BT ecosystem consisting of private and public portal sites, trackers, and torrents. Additionally, the DHTs and PEX peer discovery

mechanisms are mentioned but not considered in the remainder of the article. Although the authors claim to have discovered 8.8 million *.torrent-files* between July 25, 2008 and April 22, 2009, the number of unique info hashes in those files was only 4.6 million. The files were acquired from 5 torrent portals. With a distributed tracker crawler and 35 worker nodes, snapshots of all discovered torrents and trackers were acquired. Collecting a full snapshot was reported to take 12 hours. For the analysis covered in the article, a snapshot from April 22, 2009 was used. In that snapshot, only 1,192,303 torrents were found to have at least one peer and were considered active.

The first question investigated was the overlap of active torrents on the different portals with the result of highly active torrents being shared on most portals and the less active torrents having more distinction among the portals. The most active tracker at that time was "The Pirate Bay" which tracked almost 5 million peers. The article found that only 1% of the torrents had a swarm size larger than 100, which implies that most torrents are not very active. Accordingly, the swarm size distribution showed a very clear Zipf behavior. The authors claim that only 44% percent download multiple files at once. However, such a conclusion can not be reliably drawn from a snapshot that takes 12 hours to collect since only a few users stay online for 12 hours or more. Resolving the IP addresses geo-location revealed that most peers resided in the USA, this could be an effect of the time of day the collection was done.

To measure 4.6 million torrents was an ambitious goal, requiring a significant amount of time to collect all the peers of all swarms. A collection of peer addresses acquired over 12 hours does not reflect an accurate snapshot of a system as dynamic as BT. There will be completely different users, *i.e.,* peers, active at the beginning and the end of the collection. Therefore, it is invalid to compare individual peers between torrents. The study is of macroscopic nature, although, complementary measurement approaches, *i.e.,* DHT and PEX, are applied.

2.3.4 Deep Diving

[53] conducted a measurement study to collect IP addresses of BT swarms. The crawler described in the paper uses only a single machine to query all trackers of a torrent repeatedly. The torrents were collected from "Mininova" and "The Pirate Bay" torrent portals.

Table 2.2: Overview of measurements from [53].

Source	Torrents	#IPs	#ISPs	Time
Mininova	latest 40 k	3.9 M	10.4 k	3 snapshots in 3 w
Mininova	latest 3 k	17.4 M	10.5 k	24 × 1 h
Pirate Bay	top 600	21.9 M	11.1 k	24 × 1 h

Table 2.2 gives details on the different measurements conducted in [53]. The first one included the latest 40,000 torrents from Mininova and was executed three times with one week between measurements. The paper states that a complete snapshot took 90 minutes to complete which is a long time span for a snapshot as churn will have some effect during this period. The IPs and ISPs are per snapshot. The other two measurements cover fewer torrents but were executed hourly for a full day, and the reported numbers are per full day.

A more detailed description of the torrents investigated would be desirable since those torrents are used to model locality and inter-domain traffic. At least, the average file size of the torrents should be considered. The time dimension of the 40 k torrents measurement is very weak as it is only done once every week it does not reflect diurnal or day of week effects. The smaller two data sets can show at least diurnal patterns but are very limited in the number of torrents they cover. Furthermore, there is no indication in the paper at what exact dates and times those measurements were executed. The measurements and the resulting dataset is of the macroscopic type.

2.3.5 Exploring BitTorrent Topologies

A microscopic BT study [17] documents a method able to discover connections inside a BT swarm without instrumenting clients like the "Ono" [51] plugin. The method exploits a feature of the BT protocol termed collision, meaning that a peer will drop a connection immediately during the handshake phase if it realizes that a connection with the same peer ID is already established. The measurement system needs to discover all, or at least most peers of a swarm by repeated tracker queries. Several nodes will then start the handshake procedure with the discovered peers, learning the peer ID from the opposite peer. Knowing the peer IDs of most peers, the nodes can spoof the handshake to impersonate another peer by using its ID. Therefore, a collision, *i.e.,* a dropped connection, means that the peer, of which the ID was used, and the target have a connection.

There are other reasons for connection drops during the handshake phase, producing false positives. Those reasons are: the target peer has already the maximum of allowed active connections, peers might block the IP of the measurement node, or the target left the swarm. The first case can be detected by delaying the handshake exchange for a few milliseconds since the connection will be dropped immediately if the peer's connection pool is empty. Blocking and old peers can be filtered out later if they never showed a negative result throughout the measurement. With an instrumented client it was determined that 85% of connections are open for 400s or longer which gives a period in which a measurement should be conducted.

The detected connections between peers were used to determine if the locality is better or worse than in a random graph by calculating the average connection distance which should be equal to the average of all possible paths in a random graph. 33 swarms between 50 and 500 peers show that there is no locality present in BT. The presented measurement approach is very useful and provides detailed information about a swarm. However, the applicability is limited as the connection discovery complexity is in the Order of $O(N^2)$, rendering the method infeasible for many large swarms.

This work is an example of a microscopic measurement as detailed information about peers of a single swarm is collected by re-implementing part of the BT protocol. The most important result of this work is the confirmation that a BT swarm is equivalent to a random network, which has implications for models of BT.

2.4 P2P Locality Awareness

The motivation behind locality awareness in overlay networks is to improve service quality metrics such as delay or packet loss and inter-domain traffic, which is usually a bottleneck and incurs a cost to ISPs. Therefore, it is beneficial to increase locality in BT swarms, which can, in turn, improve performance for the users [8]. In the context of overlay networks, locality does not necessarily translate to physical closeness. Locality can also be understood as closeness concerning network hops or delay. The reason for this semantic duality is that hop count and physical closeness are not correlated on a microscopic scale, *i.e.,* a network connection between physical neighbors can run through another city if the neighbors use different ISPs. However, on a macroscopic scale, physically close users, *i.e.,* from the same country or even continent, are typically also close in networking terms compared.

2.4.1 Biased Neighbor Selection

Biased Neighbor Selection (BNS) [4] reduces cross-ISP, or inter-domain, traffic. The approach is to connect peers that are in the same ISP with a few exceptions to prevent fragmentation of the swarm. A solution like this can be implemented on tracker or peer level with the help of IP to AS maps. It is possible for ISPs to intercept and modify tracker responses to include more local peers.

BNS was evaluated by simulating 14 ISPs with 50 peers each which had a homogenous asymmetric bandwidth of 100 kbps upload and 1 Mbps download respectively. ISPs were connected in a full mesh topology together with some university nodes that had symmetrical bandwidths. The results show that BNS reduces variation in download times and reduces traffic redundancy, traffic repeatedly coming into an ISP's domain.

2.4.2 Biased Peer Selection

The "Ono" plugin was developed for the Vuze BT client, formerly known as Azureus, introduced by [8]. The goal of "Ono" is to improve the BT topology such that peers close to each other prefer downloading from each other, this approach is also called biased peer selection, which is a variation of the BNS approach discussed before.

To determine proximity of peers "Ono" exploits Domain Name System (DNS) resolution of large Content Delivery Networks (CDN), since those already collected information on network topology to assign requests to the closest server. The argument is that close peers will be redirected to the same CDN subnet. After comparing the cosine similarity of each peer's maps, mapping CDN hostnames and subnets, the most similar peers are preferred. The authors claim that "Ono" can execute DNS lookups on behalf of peers that do not have the plugin. However, the authors did not explain how this is done, since DNS lookups on behalf of others is not an officially supported feature of DNS. Besides DNS lookups, also traceroutes and AS hops are collected, it is likely that those data was used to bias neighbor selection of non-"Ono" peers.

The results show that AS hops can be reduced by 50% and many paths can be found to peers inside the same AS. Examples revealed that ISPs that allocate more bandwidth to intra-domain traffic increase the performance BT in conjunction with

"Ono". Finally, CDN domain name resolution proved to be a feasible oracle for proximity between peers.

2.4.3 Biased Unchoking

Biased Unchoking (BU) [47] is an optimization of BNS on a lower layer of the BT protocol. The choking algorithm orchestrates to which peers chunks are uploaded. It works in a round robin fashion where every 30 seconds the neighbor peer which uploaded the most is unchoked, meaning uploading starts. The neighbor that was unchoked the longest is again choked. This algorithm enforces the "tit-for-tat" incentive scheme in BT, meaning that peers, which upload more, can also download more [13]. Seeders select peers to unchoke randomly as no upload is present, this is called "optimistic unchoking".

BU is applied on seeders to unchoke local peers more often than remote ones. Thus, an oracle service, which gives a closeness score for two IP addresses, is needed. For the simulator in [47] the AS hop distance was used with a threshold of zero, meaning that only peers in the same AS are preferred and are also unchoked as long as preferred peers are present.

A simulation was used to compare standard BT with BNS, BU, and BNS combined with BU. The results show that BU and BNS work very well together and can greatly reduce inter-domain traffic if the network is under high load. The combination of BNS and BU is especially useful in scenarios that have bottlenecks on the inter-domain links as the performance of the BT network can be increased. However, the simulation seems to ignore the fact that peers can unchoke more peers if their bandwidth is not entirely utilized which would possibly also improve the performance in the constrained inter-domain link scenario. Finally, investigations on how BU affects the "tit-for-tat" incentive and free riding are yet missing.

2.4.4 Neighbor Suggestion

Neighbor Suggestion was first described in [17] and is based on PEX since it can be applied without changing client software. A special agent is opening BT connections to peers in a swarm and uses PEX messages to suggest new peers to its connected peers. Those suggestions are biased to provide as many local peers, as there are available from the tracker, increasing the chances of local connections being made. The implementation was done in similar fashion as the measurement approach described in Section 2.3.5 having one coordinator and several worker nodes which make the connections.

The evaluations conducted used a measurement method (*cf.* Section 2.3.5), allowing to show the impact of applying Neighbor Suggestion to BT swarms. The presented results from 79 swarms indicate that on average locality is improved by roughly 6%. The improvement is small while the solution is quite difficult to deploy on a large scale capable of managing all existing swarms. However, the work showed that potential for the locality in optimizations in BT still exists and that formerly proposed solutions did not find their way into practice.

The survey of literature presented herein covers the field of BT measurements and locality mechanisms. With the relevant related work reviewed, the current state of research is summarized and an existing gap in BT is identified.

2.5 Gap Analysis

A selection of the most significant measurement studies was described and categorized. To provide an overview of those macroscopic studies, Table 2.3 presents the covered studies and their key metrics. For each dataset presented in related work, the most extensive representative is present. For some of those data sets the information provided in the corresponding literature was incomplete, often the time it took to gather a snapshot, denoted ΔTime, is missing. The last dataset, Blind [51], does not fit the comparison very well as it collected different data than the others and can not be clearly put in one measurement category. However, it was included for completeness. The VIOLA 2016 dataset (*cf.* Section 4.2.1) is included here to show that it fills the gap in BT measurements.

Table 2.3 summarizes the key parameters of the previously presented macroscopic BT measurements. The type column defines how the measurement was executed the options are portal crawling, tracker crawling, distributed measurement, PEX crawling, or the "Ono" plugin. Samples indicate how many samples were taken during the whole measurement and ΔTime is the total time taken for the measurement. Detail is related to the level of detail covered by the measurement, *i.e.,* seeder and leecher numbers (S/L) found on portals and trackers or IP addresses. The last column, #Swarms, indicates the number of swarms covered.

Table 2.3: Comparison of key attributes of macroscopic BT data sets.

Data Set	Type	Detail	Samples	ΔTime	# Swarms
TV. [24]	Portal	S/L	96	36 h	63,867
Pop. [24]	Tracker	S/L	1	n/a	4,463
Grp. [24]	Distributed	IP	440	88 h	16
Mus. [24]	Distributed	IP	1	n/a	135,679
Zhang [76]	Distributed	IP	1	12 h	1,192,303
mn40k [53]	Tracker/PEX	IP	1	90 m	40,000
Blind [51]	Ono	IP	n/a	1+ y	n/a
VIOLA 2016	Distributed	IP	6,624	2,208 h	70,291

The table is visualized as a bubble chart in Figure 2.7, which presents the number of swarms covered by the dataset on the y-axis on a log scale and the samples per hour on the x-axis. The size of the bubbles shows the level of detail; small bubbles consist only of seeder and leecher numbers whereas the larger ones also contain IP addresses of the peers. Not available values, depicted "n/a" in the table, were replaced with 1. The Blind [51] dataset is, thus, counted as having one swarm due to the lack of identification of swarms in the data, although more than one swarm was measured. The other bubbles gather almost on a line, reflecting the trade-off between covering a high number of swarms and a low ΔTime per sample. Either many swarms can be covered, or many samples can be repeatedly taken within a short period of time. However, before the VIOLA 2016 measurement, there was a gap at around 100 to 10,000 swarms with 2 to 4.5 samples per hour. Under the aspect of the Pareto effects of peer distribution, *i.e.,* 20% of swarms have 80% of the peers this is the most interesting area to conduct a measurement since only the large swarms are of concern. Thus VIOLA is covering this gap.

The key aspect, which received little attention in previous work, is the measurement methodology itself. Although reasonable assumptions were typically made, they were not investigated closely. The largest gap here is in the time dynamics of swarms in the short term. It remains to be investigated how accurate a measurement is that takes a minute or more since a swarm is constantly changing. Furthermore, the question of how to acquire a fully accurate snapshot is to be answered, or a quantification of the error should this not be possible. A further aspect that was not considered in most measurements is DHT tracking which has become more and more important, especially since the introduction of magnet links which work without a

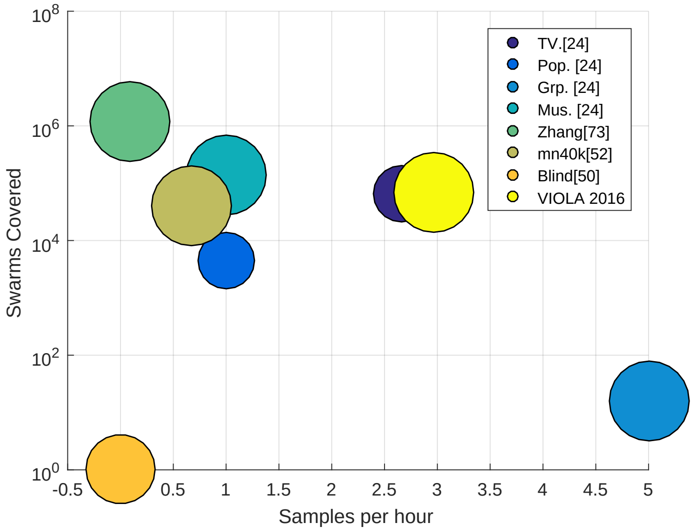


Figure 2.7: Comparison of prior BT measurements.

tracker meaning that some peers might not use trackers at all. Those measurements potential miss a significant part of the system.

Existing studies [24, 17] confirm that there is no location bias in the BT network although several mechanisms have been proposed. The "Ono" [8] plugin was one of the few locality awareness mechanisms that were officially available to users. However, with the shrinking popularity of the Vuze client, the plugin loses relevance. All data sets described are a few years old and, thus, it is worth investigating if the random distribution is still observable. There is still little to no research on content popularity under the aspect of locality and no long-term data to investigate these issues is available. The question of what characteristics of content define where it will be consumed most is only partially answered. There are indications that language has an influence but is not the sole factor defining the locality of content. However, such knowledge would be beneficial for multiple stakeholders, *e.g.*, for ISPs to optimize caching and prefetching, for content providers to define which content to make available in which regions, or even for P2P users who can benefit from performance gains.

Based on these issues discussed, the following areas are identified which are prime research targets to provide scientific contributions:

1. BT measurement methodology in general and specifically regarding DHT tracking.

2. Long-term observation of swarms with multiple samples per hour over a period of months.

3. Analytic methods and their results to investigate important countries and ASes as well as the effects of content popularity.

Therefore, Chapters 3 to 5 provide detailed descriptions of those contributions. Specifically, Chapter 4 revisits that gap analysis presented in Figure 2.7 to show how the measurements system presented in this thesis can close that gap.

Measurement is the first step that leads to control and eventually to improvement. If you can't measure something, you can't understand it. If you can't understand it, you can't control it. If you can't control it, you can't improve it.

H. James Harrington

3

VIOLA – Measuring BitTorrent

Collecting all peers of a swarm is challenging as the system does not – by design – support such a collection. Thus, an analytical analysis of the problem of collecting all peers from a swarm is required. Based on this analysis, the design and implementation of a measurement system to close the gap in BitTorrent (BT) measurements (*cf.* Section 2.5) are presented.

To collect a complete swarm one needs to know how big the swarm is, *i.e.,* to determine when the collection is complete. Due to BT's distributed architecture with many trackers and two Distributed Hash Tables (DHT) and due to the abundance of client software and index sites available, it is very unlikely that one entity knows all peers being active at any point in time. Additionally, measurements involving trackers or DHTs take time, during which the state of a swarm is changing due to churn. Thus, churn might lead to inaccurate measurements and needs to be investigated. Since the collection of all peers from one or multiple trackers is a modified version of the Coupon Collector Problem [75], the newly termed BitTorrent Peer Collector (BTPC) is introduced here. This new approach contributes and evaluates a novel method to estimate swarm sizes based on tracker or DHT responses. For that evaluation, a dataset was collected, containing tracker and Distributed Hash Table

Table 3.1: Notation used for the remainder of this thesis.

Symbol	Description
N	Real swarm size, *i.e.,* ground truth
N^*	Estimated swarm size
k	Number of peers in a response
k_{rel}	Response size relative to swarm size
Δk	Time required for one request
$\mathcal{M}$	Number of unique peers collected
$\mathcal{M}^*$	Predicted unique peers collected
Y	Number of total peers collected
i	Number of queries
λ	The join rate of peers per second

(DHT) responses for one swarm over 24 hours. Based on this dataset an analysis of the impact of churn on measurements and estimations is performed.

This measurement and its validation provide the basis for the design and implementation decisions presented in Section 3.3 to design the VIOLA measurement system which can fill the existing gap in BT measurements. The requirements of such a system are elicited before a scalable architecture is presented. Finally, the unique design decisions which were required to build the VIOLA system are presented. Table 3.1 presents the notation used in this chapter.

3.1 The BitTorrent Peer Collector

The BitTorrent Peer Collector (BTPC) [37] problem is a combination of the Coupon Collector (CC) and Reverse Coupon Collector (RCC) problems as a swarm's size is unknown and tracker responses are random. The original CC can be stated like:

> Given that there are N different coupons available in boxes of a certain product, what is the probability that after buying $\mathcal{M}$ such boxes, one will have collected exactly i different coupons? [32]

Applied to BT this means that N is the swarm size, the boxes are the tracker responses, and $\mathcal{M}$ is the number of received peer addresses. However, there is a difference. Tracker or DHT responses contain multiple distinct peer addresses, unlike the single coupon in a box. Although trackers include the number of seeders and leechers of a swarm within their responses, a tracker only knows the peers that re-

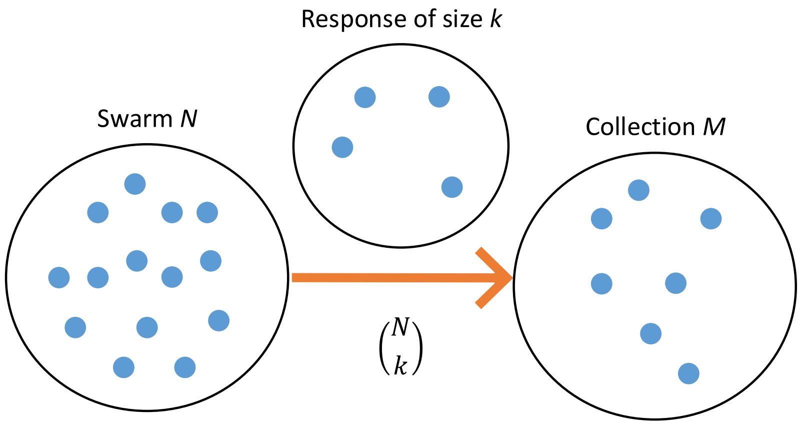


Figure 3.1: Illustration of the BTPC.

gistered with it. Since there are typically multiple trackers used for one torrent, it cannot be assumed that one tracker knows all peers in a swarm. Furthermore, some peers might not even use a tracker and rely solely on the DHTs for peer discovery. Therefore, to answer the question the only option is to estimate how many peers are in a swarm,*i.e.,* the RCC:

> For fixed i, M, what value of N maximizes the probability $p(i, M; N)$? That is, given i, M, what is the most likely value of N in the Maximum Likelihood sense? [32]

The Maximum Likelihood Estimator (MLE) for the RCC [32] can be used to answer how many peers need to be collected from trackers and DHTs to collect the whole swarm. The difference to the classical Coupon Collector is the response size which is 1 in the standard CC as opposed to the case of BT, where the response size is typically 50 or larger and can be heterogeneous. In the general Coupon Collector [75] problem the goal is to find all coupons from a set of coupons by drawing one coupon at a time randomly where the distribution of coupons is not uniform, and therefore the probability of drawing a coupon depends on the type of coupon.

To collect all IP addresses – coupons – of a swarm, the collector has to query a tracker to receive a set of IP addresses – draw – until all addresses are collected. Figure 3.1 illustrates such a draw, k, which is randomly chosen out of N, *i.e.,* a k-

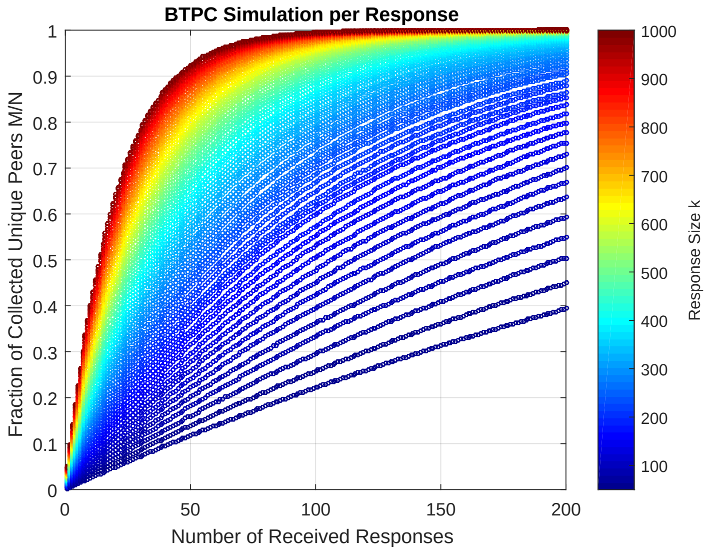
combination of N written as $\binom{N}{k}$ (read as "N choose k"). By selecting k unique addresses from N, tracker and DHT requests are modeled based on the BT design [13] that the responses are uniformly distributed. For the remainder of this thesis, the number of unique peers collected after i draws shall be denoted M_i, the duplicates contained within response i by d_i, and the total number of peers collected Y_i. Table 3.1 summarizes the notation used in this thesis.

Based on those properties of the BT system a simulation was created to illustrate the effects of the BTPC. Figure 3.2a shows the portion of the swarm discovered after X responses of size k have been received where k is varied from 50 to 1,000. As expected, with larger response sizes more unique peers are discovered than with the same amount of smaller responses. More importantly, the shapes of the curves indicate that the fraction of unique collected peers asymptotically approaches 100%. Due to those random responses, the probability of new addresses being found in a response gets smaller, the closer M approaches N. Therefore, it will be very difficult to collect all peers of a large swarm. However, discovering a major part of a swarm, *e.g.,* 95%, seems to be feasible.

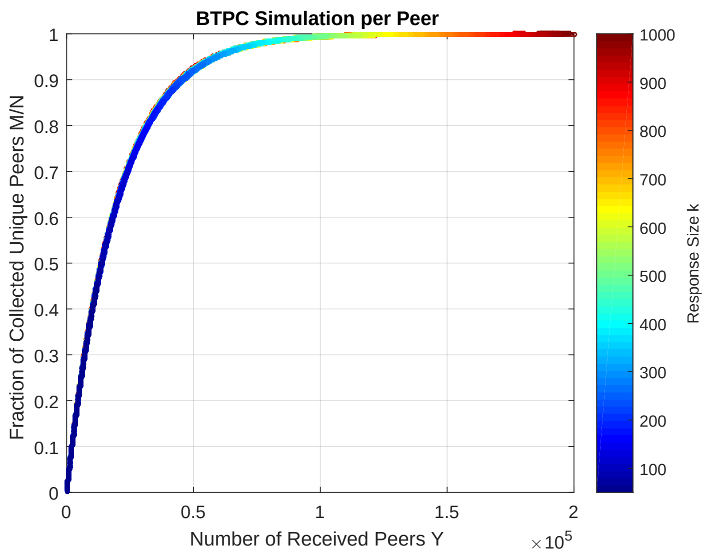
Figure 3.2b presents the same simulation data as Figure 3.2a, the only difference is the x-axis, which has been changed to show returned peers. While Figure 3.2a showed the number of responses received, Figure 3.2b shows the cumulative sum of peers returned by all the responses, or $k \cdot \#responses$. The fact that all the points lie on the same trajectory indicates that the size of a response k does not influence the number of unique peers found for the number of peers received. Thus, if k is significantly smaller than N, k does not have a measurable influence on the discovery rate of unique peers. This leads to the conclusion that treating every peer address received as a single observation will not affect the accuracy of the MLE. Under those conditions the BTPC can be formulated:

> With unknown swarm size N and average response size k how many queries i have to be made to collect every address in a swarm?

This problem defers from the general CC [32, 75] in the unknown N. Thus, the RCC has to be solved before the CC. In practice, this will be an iterative process since with every response received the collection M grows and the swarm size estimation N^* can be improved. Since the MLE is computationally intensive, the simple estimation approach is developed before going into detail on how the MLE



(a) After X responses have been received.



(b) After X peers have been responded.

Figure 3.2: Portion of the swarm of size $N = 20,000$ discovered per requests and per peers.

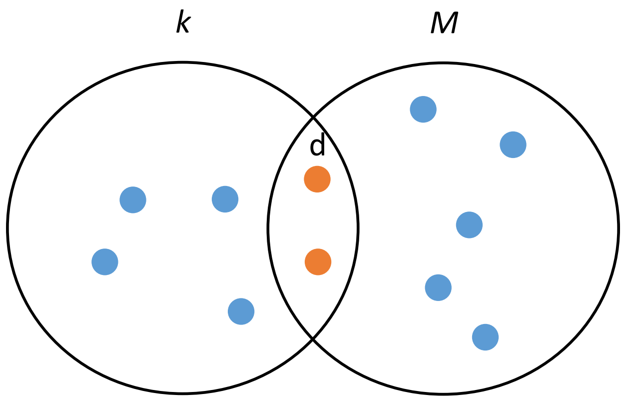


Figure 3.3: Illustration of the duplicates contained in tracker responses.

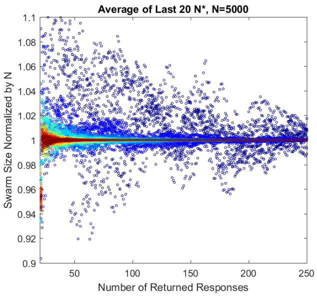
can be applied to BT. Both methods are compared based on a time series of tracker- and DHT-responses from a real BT swarm.

3.1.1 Simple Estimation

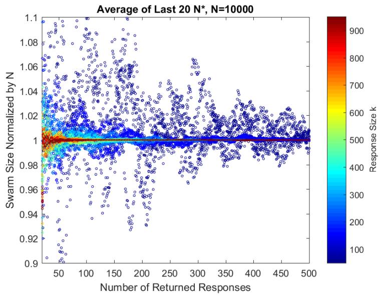
The first approach for estimating the swarm size involves counting duplicates contained in responses. Figure 3.3 illustrates those duplicates, being part of the response and of the collected peers M. Since the peers in the response are uniformly randomly distributed the ratio of duplicates to response size is, on average, the same as the ratio of discovered peers to swarm size, *i.e.,* $\frac{M_{i-1}}{N_i^*} = \frac{d_i}{k}$. Therefore, an estimation N_{simple}^* can be made with each response after the second response is received (because $M_0 = 0$ and $d_1 = 0$) by solving for N_{simple}^* as in Equ. 3.1.

$$N_i^* = M_{i-1}\frac{k}{d_i} \tag{3.1}$$

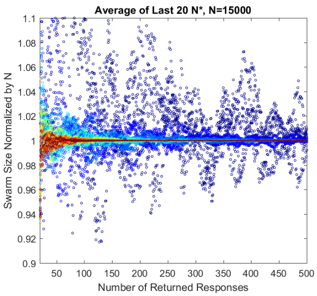
Figure 3.4 presents simulation results for swarm sizes between $N = 5,000$ and $N = 100,000$ providing a broad view on the effect of swarm size on N_{simple}^*. The response size was varied between $k = 50$ and $k = 1,000$ in steps of 100 and each k was run four times because plotting more than four runs renders the figure unreadable. The number of responses plotted is proportional to N and, thus, the higher N, the denser the plot appears. N_{simple}^* has been normalized by the swarm size N to cen-



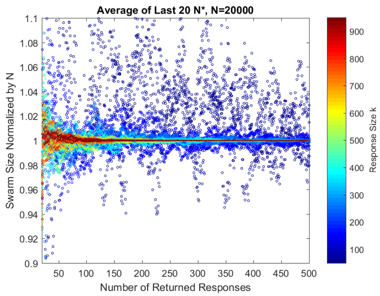
(a) N = 5,000



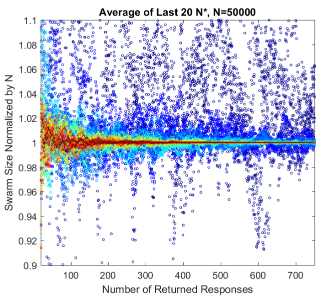
(b) N = 10,000



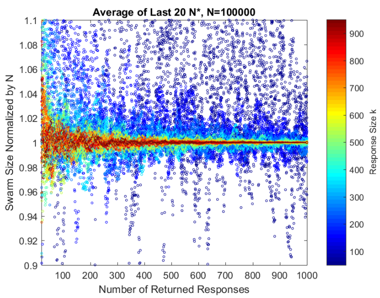
(c) N = 15,000



(d) N = 20,000



(e) N = 50,000



(f) N = 100,000

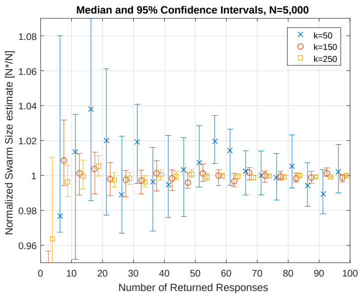
Figure 3.4: Simple swarm size estimations for swarm sizes between 5,000 and 100,000.

ter the plots around one, which also represents the actual swarm size N. To receive better and more consistent results, the moving average of the last 20 N^{*}_{simple} was taken. For this reason, the plots start at 20 responses. The first observation is that the accuracy increases with increasing response size and increasing number of responses; visible by the blue dots that are scattered around while the red dots are closer to one. This is explained by the higher probability of collecting duplicates with higher k and larger M, meaning that the estimate gets more accurate with larger ks. The second observation is that N^{*}_{simple} tends to overestimate the swarm size. This effect seems to be more pronounced the larger N is.

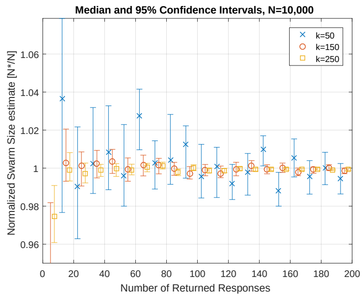
To be able to make statistically sound statements more than four simulation runs are required. Thus Figure 3.5 shows the median and the 95% confidence intervals for 100 runs of the simulation for different swarm sizes. With 100 runs the confidence intervals are getting reasonably small to prove the effects indicated in the scatter plots. Only the three smallest ks are shown since those are the least accurate and illustrate the effect of k on the estimation well. The statistical analysis confirms that N^{*}_{simple} generally overestimates N, but converges to N with increasing number of responses received. Furthermore, the estimate converges faster with larger response size. Even with the smallest k reasonable estimates can be made for the $N = 100,000$ swarm even after 500 requests have been made. Amounting to Y being one-quarter of N, leading to the rule of thumb that $\frac{N}{4}$ addresses need to be received for the simple estimate to come into the 2% range of the actual swarm size. Some of the graphs show a rising estimate for up to the first 200 responses received. This observation is explained by the absence of any duplicates in some of those first replies, meaning that N^{*}_{simple} cannot be calculated due to a division by zero (*cf.* Equation 3.1). In those cases, the simulation considered N^{*}_{simple} to be zero since no calculation can be made. The over-estimation is caused by the number of duplicates per response which is normally distributed due to the uniform probability distribution of receiving addresses. Since d is used in the denominator of Equ. 3.1, ds lower than the average have a stronger impact on the result than those higher than average. Using the median instead of the mean reduces this effect.

3.1.2 Maximum Likelihood Estimation

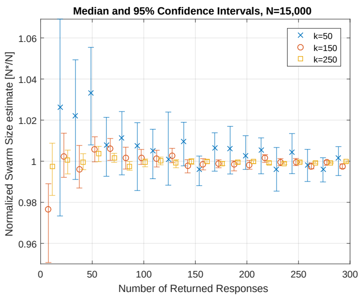
A more general and accurate solution to swarm size estimation can be achieved with a Maximum Likelihood Estimator (MLE). An MLE calculates the probability at each



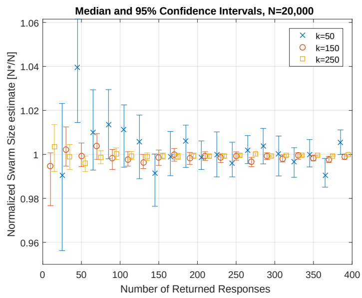
(a) N = 5,000



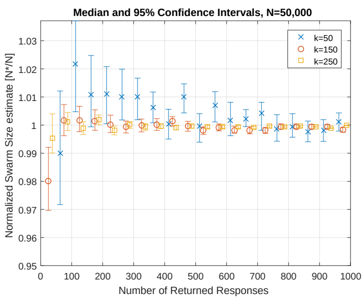
(b) N = 10,000



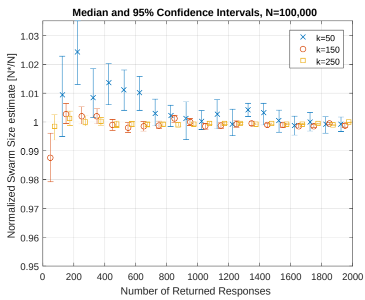
(c) N = 15,000



(d) N = 20,000



(e) N = 50,000



(f) N = 100,000

Figure 3.5: Swarm size estimates median and 95% confidence intervals from 100 runs.

step i for a range of possible swarm sizes N, the swarm size with the highest probability will become the estimate N^*_{MLE}. This way the response size k can be ignored, and each returned Internet Protocol (IP) address is treated as a single observation equivalent to $k = 1$. Therefore, a sequence of peers x is observed, *e.g.,* $x = [1, 2, 3, 4]$. The probability, q_i, to observe a new i-th peer is the number of undiscovered peers divided by the swarm size N:$q_i = \frac{N-M_{i-1}}{N}$. Vice-versa the probability, $1 - q_i$, to observe a duplicate i-th peer is the number of discovered peers divided by the swarm size N: $1 - q_i = \frac{M_{i-1}}{N}$ Using both formulas, the probability $P(N|x)$ to observe x for a given N can be expressed like:

$$P(N|x) = \prod_{i=1}^{Y} p_i \tag{3.2}$$

$$with \quad p_i = \begin{cases} q_i, & \text{if i-th peer is new} \\ 1 - q_i, & \text{if i-th peer is duplicate} \end{cases}$$

It is $Y = |x|$ the total number of returned peers.

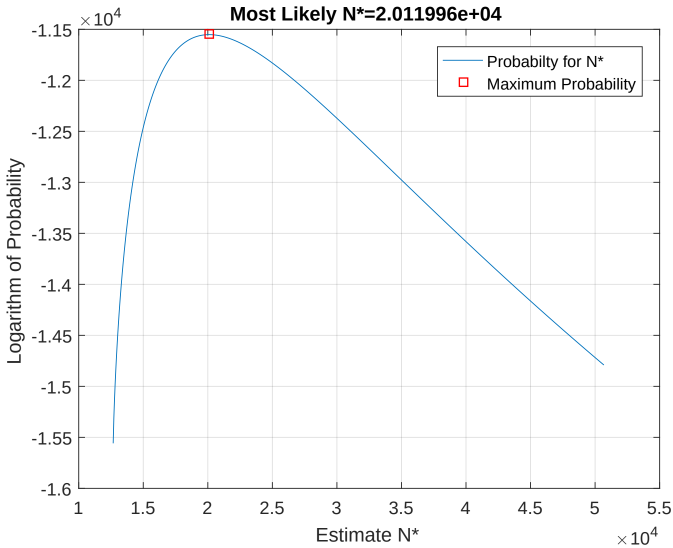
It is $M_0 = 0$ and $M_1 = 1$

Equ. 3.2 can be used as an MLE by finding the maximum probability $P(N|x)$ for a pattern x indicating the most likely estimate N^*_{MLE}:

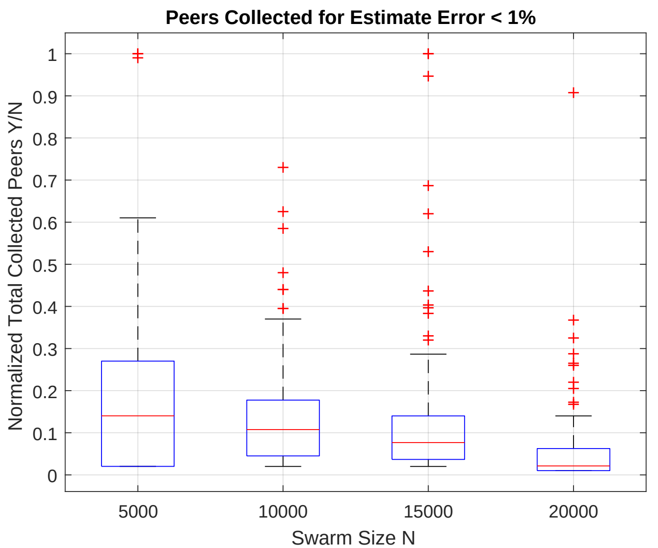
$$N^* = \max_{N=|M|:\infty} P(N|x) \tag{3.3}$$

At any point after Y peers have been seen, possibilities for all N larger than M_i are calculated. The N resulting in the largest probability is selected as the most likely N and denoted N^*_{MLE}.

Figure 3.6b shows a comparison of MLE swarm size estimation results which are in the range of $\pm 1\%$ of the real N. For numerical reasons, the log-likelihood is used to give a more robust numerical evaluation, *e.g.,* for large Ns. The box plot shows median, 25th, and 75th percentile, and the outliers of the number of peers collected, Y, divided by the swarm size, N, of the first values that estimated $N^* = \pm 1\%$ of N. First, the plot shows that for larger swarms a smaller fraction of collected peers Y is required to get an accurate measurement. This observation implies that the MLE is more dependent on collected peers Y than on swarm size N. As a general



(a) Illustration of an MLE comparison of N^*.



(b) MLE comparison with boxes showing median, 25th and 75th percentile, and outliers.

Figure 3.6: Illustration of MLE and statistical analysis of accuracy with varying swarm sizes.

rule, 4000 peers or more need to be collected to get accurate estimates. In a practical implementation, the accuracy also depends on the range and resolution of the Ns selected to calculate the probabilities.

3.1.3 Analytical Collector

An analytical solution to the BTPC problem is preferable as it can be used without much overhead to decide when to stop querying a tracker or to determine how many queries have to be executed to collect a certain fraction of a swarm. k can also be expressed relative to N like $k_{rel} = k/N$, which is the fraction of the swarm returned for each request. As N is always bigger or equal to k and k is not zero the range of k_{rel} is $(0, 1]$. Thus, with each response i the pool of collected peers M_{i-1} grows by the newly collected peers $N - M_{i-1}$ times the relative response size, *i.e.,* Equ. 3.4.

$$M_i^* = M_{i-1} + (N - M_{i-1}) \cdot k_{rel} \tag{3.4}$$

This formula is simpler than using simulation data but still not elegant since it is recursive and, thus, hard to compute for large is. To simplify things one can look at the number of a swarm's not collected peers which will decrease with the rate $r = 1 - k_{rel}$. With each response received the the number of unknown peers decreases as in Equ. 3.5.

$$N - M_i = N \cdot r^i \tag{3.5}$$

To obtain the number of discovered peers the expression can be subtracted from 1 and r can be substituted with $1 - k_{rel}$ which gives the formula for the fraction of peers found after the i-th response of size k has been received, *i.e.,* Equ. 3.6.

$$M_{rel}^* = 1 - (1 - k_{rel})^x \tag{3.6}$$

This formula produces a result relative to N. If an absolute number is desired, the result has to be multiplied by N. The calculated M^* can be compared to the simulated M to determine the goodness of fit of the model by calculating the coefficient of determination (R^2). Applying R-squared to a simulation with 50 ks ranging from 1 to 1,000 the mean of R^2 is $\bar{R^2} \approx 0.999987$ for all the ks simulated. That is an excellent fit and means that 99.9987% of the variance in the model can be explained by the model. This model can also be applied to multiple collectors by replacing x with

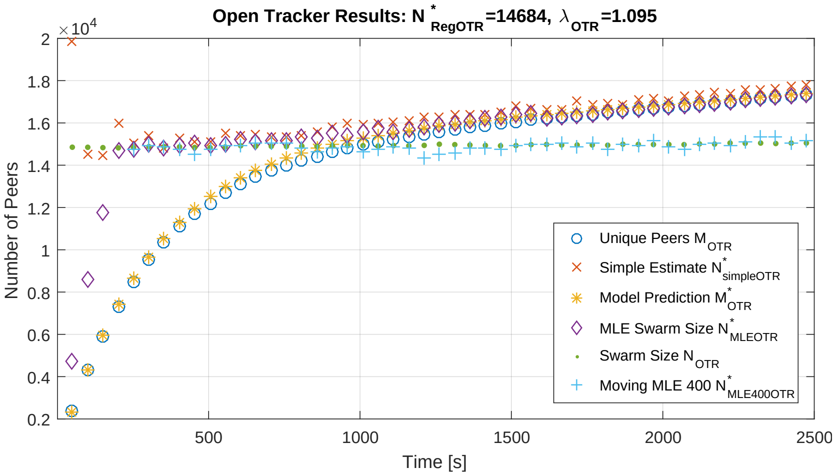


Figure 3.7: The first 2,500 s of measurements for OTR.

the number of collectors and replacing k_{rel} with the average M_{rel} of those collectors, assuming that all collectors have comparable values of M_{rel}.

3.2 Validation

To validate those concepts established in Section 3.1 and to investigate the impact of churn, a measurement was collected. Those measurements were done for one torrent over 25 hours from May 9, 2016, 08:05 GMT. The data was acquired by sending one announce request per second, $\Delta k = 1$, to each of the four responding trackers in the torrent "Deadpool" [29] and the mainline DHT. This torrent was chosen because it ranked highest in the movie category on the "Kickass Torrents" [30] portal at the start of the measurement. The data contains a timestamp, IP addresses, seeder, leecher, and total peer count data for tracker responses. The dataset is available for download at [36]. For ethical reasons IP addresses contained in the dataset were anonymized to prevent the identification of individual users while still maintaining the uniqueness of IP addresses for analysis.

3.2.1 Quantifying Churn

A BT measurement over a period, such as those 25 hours, will inevitably be biased by churn. Thus, it is of vital importance to quantify that bias for the conducted measurement. To evaluate the tracker case, responses from the tracker with the largest swarm size during the measurement period are used. In this instance, this was the "Open Trackr" (OTR) [48] which initially reported a swarm size of $14,877$. Figure 3.7 depicts the first 2,500 OTR replies, which equals the first 2,500 s. The circles show M_{OTR}, the unique peers collected from OTR. At first glance, the pattern seems to be as expected from the simulations in Section 3.1. However, the circles surpass the swarm size announced by the tracker N_{OTR}, what should not be possible since collecting more unique-peers than the swarm size is not logical. This effect is explained by peers constantly joining and leaving the swarm. Thus, M_{OTR} will contain peers that have already left. With the current measurement, it is not possible to filter those peers as very accurate snapshots of the swarm would be required. The crosses represent the simple estimates, $N^*_{simpleOTR}$, and the diamonds the MLE, $N^*_{simpleMLE}$. After being close to the reported swarm size between 250 s and 500 s, both follow a curve close to M_{OTR}.

The increase of M_{OTR} and both estimates is a result of peers joining the swarm over time, which will be collected and considered to be new unique peers. Leaving peers that were already collected do not influence this result, only those leaving before being collected might slightly reduce estimates in the beginning since M_{OTR} is slightly smaller than expected. Therefore, for measurements, the main concern are the peers joining per second, termed the join rate and denoted λ. Since the increase of M_{OTR} is almost linear for $3,000s < t < 5,000s$ and λ_{OTR} is approximately constant, linear regression can be applied to estimate λ_{OTR}, *i.e.,* the slope of the curve. The y-intercept of the regression gives another estimate of the swarm size N^*_{RegOTR} at $t = 0$. In this case, the slope was $\lambda_{OTR} = 1.095$, meaning that 1.095 peers join the swarm per second. The intercept was at $N^*_m RegOTR = 14,684$, meaning that there were $\approx 14,700$ peers in the swarm at the start of the measurement, being very close to the swarm size reported by the tracker $N_{OTR} = 14,877$ being 1.3% off. Finally, the join rate λ can be included in the model from Equ. 3.6 to analytically calculate M^*_t with Equ. 3.7, the unique peers at time t. The result of applying the revised model is shown in Figure 3.7 as asterisks. It reaches an accurate match to M_{OTR}, although not a perfect one. Note, that this model is time dependent as λ is time dependent.

$$M^*_t = (\lambda t + N) \cdot (1 - (1 - \frac{k}{N})^{\frac{t}{\Delta k}}) \tag{3.7}$$

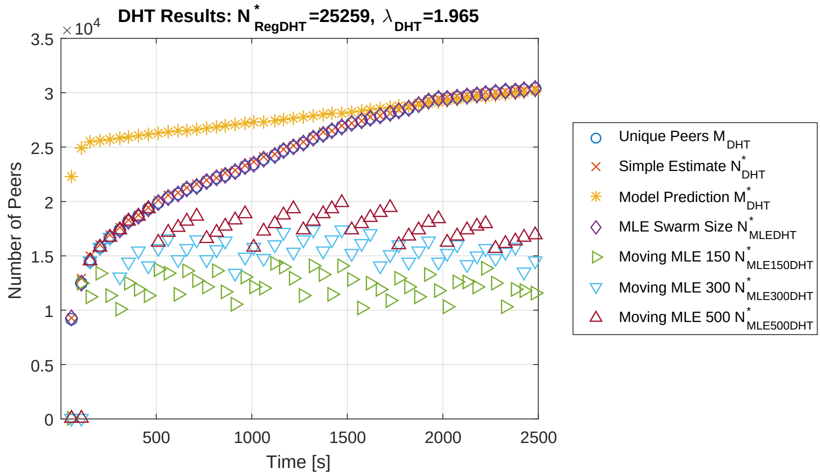


Figure 3.8: The first 2,500 s of measurements for Mainline DHT.

These MLE results are accurate for a period of time Δt in which churn does not have a noticeable influence on N^*_{MLEOTR}. Section 3.1 revealed that with 30% of N collected, *i.e.*, $Y \geq N \cdot 30\%$, in 75% of the cases MLE estimates are accurate to 1%. Therefore, accurate estimates can be expected for $Y \geq 5{,}000 peers$ which translates to 100 received responses or $\Delta t_{min} = 100s$ of measurements. Figure 3.7 shows $N^*_{MLE400OTR}$ for $200s < \Delta t < 400s$ is plotted, twice Δt_{min} to receive smoother results and four times Δt_{min} as the upper bound to reduce calculation overhead. These $N^*_{MLE400OTR}$ results follow N_{OTR} with very small deviation. The relative error E_{M_i} introduced by λ_{OTR} for $t_{max} = 400$ can be calculated by subtracting the predicted value without churn $M{*}_{OTR400}$, Equ. 3.6, from the real M_{OTR_400} and dividing by M_{OTR400}, which amounts to $E_{M_{OTR_400}} = 1\%$. Based on Equ. 3.7, the relative error between our measurements and the model can be derived, allowing to derive the required time span Δt such that the relative error is $< \varepsilon$ with probability p for given λ. The main problem is that the churn rate λ needs to be accurately determined. This is, however, beyond the scope of the manuscript and left for future work, as measurement series need to be conducted to validate the simple churn estimator.

Figure 3.8 shows the respective DHT results, lacking the swarm size due to the fact that this information is not available in the DHT. As expected, due to the larger response size, M_{DHT} increases faster than M_{OTR} in the beginning, but also the

$\lambda_{DHT} = 1.965$ and $N*_{RegDHT} = 25,259$ estimated by linear regression are higher. Thus, more peers use the DHT than the OTR and over-proportionally more peers join the DHT than the OTR because there is only one official DHT, while there are multiple trackers, *i.e.,* 4 in this case. Additionally, censorship and Internet blockades can have an influence, as it is much more difficult to block a DHT than blocking certain tracker addresses. An effort of fitting the model from Equ. 3.7 to the DHT results in a bad fit to the actual $\mathcal{M}_{DHT}$ as the discovery of unique peers is not as fast as expected. This is due to different behavior of DHT responses compared to trackers, based on different timings in Mainline Distributed Hash Table (MDHT) client and tracker code. Accordingly, moving MLEs do not work for the DHT case. Figure 3.8 shows MLEs calculated in the same fashion as for the OTR data. The larger Δt, the larger the estimate becomes. The reasons are the MDHT implementations [43] in BT clients. One possible practical solution is to use multiple measurement nodes in parallel to reduce the time and, thus, the effects of churn on peer collection.

The presented results of this section show that the BTPC Problem introduces an error into measurements which can be quantified by comparing the theoretical model in Equ. 3.6 to the measured $\mathcal{M}$. With the evaluated models, it becomes possible to estimate swarm size and calculate the number of requests required to collect the swarm. If those requests are executed in parallel, the time required to collect the required samples for an accurate estimate is greatly reduced and churn can be neglected. Those investigations of the BT system form the basis for a measurement system design, which can close the gap in BT measurements.

Those investigations lead to the conclusion that the simple estimate is well suited for swarm size estimation for collection. Compared to the MLE it overestimates swarm size with only a few responses, which, in the worst case, leads to over collection. However, as the estimate is not feasible with a DHT, its main use is to confirm tracker reports. To develop an estimator for the DHT the exact implementation needs to be investigated and included in the model. More important for the implementation of a distributed, measurement system is the analytical collector model (*cf.* Equ. 3.6), which is the key to coordinate distributed swarm collection with minimal overhead. These investigations constitute the basis for the Video Consumption in Overlay Networks (VIOLA) design.

3.3 Systems Design

The VIOLA measurement system was designed based on the findings documented in Section 3.1. This section details how the models and equations can be applied in a concrete system implementation. For this, a top-down approach of explaining is used, explaining first the goal and requirements for VIOLA, continuing with the resulting architecture and, finally, detailing implementation aspects applied to realize the design.

3.3.1 Design Goals

In contrast to existing BitTorrent measurement systems (*cf.* Chapter 2), which typically take snap-shots of the overlay network, VIOLA can monitor a large number of swarms over time to fill the existing gap in BT measurements. Filling this gap means collecting $\approx 10,000$ or more swarms every three times per hour over a period of months. For this purpose, the primary design goals are summarized in the following.

1. Flexibility: VIOLA can be used in different scenarios, *e.g.,* to monitor different categories of torrents because the category or selection of torrents to be measured is typically not known beforehand and has to be changed if BT changes.

2. Scalability: Due to the flexibility, swarm number and swarm sizes are not known previously. Thus, VIOLA needs to be horizontally scalable to be able to cope with high loads, which likely occur when collecting 10,000 swarms in 20 minutes.

3. Autonomy: Since torrents have a limited lifetime, it is not feasible to select the range of torrents to be monitored at the start of a measurement. Therefore, VIOLA needs to be capable of discovering newly released torrents without the interaction of an administrator.

4. Compatibility: The output of the system needs to be easily accessible with standardized methods and tools to facilitate analysis and use of the collected data.

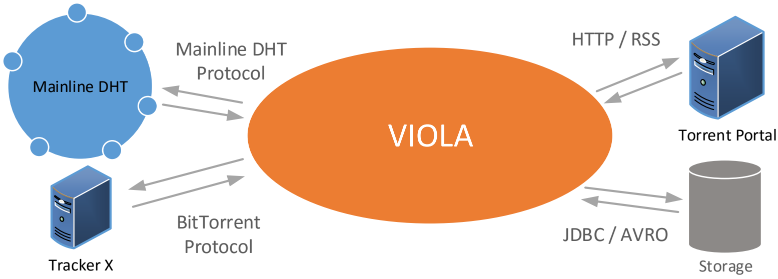


Figure 3.9: Scope of VIOLA with interfacing systems.

5. Raw Data: Ideally, all measurement data and metadata is kept by a measurement system to guarantee reproducibility. Therefore, the retained data needs to reflect the relevant measurement metadata, such as which addresses were contained in which announce response, as well as the actual data.

Those high-level goals ensure that VIOLA can reach its main purpose of contributing a novel BT dataset and that it can be re-used in other scenarios.

3.3.2 Architecture

Figure 3.9 depicts the scope of the VIOLA measurement system and the systems with which VIOLA interacts. The existing components of the BT network are depicted on the left side of VIOLA. VIOLA implements the necessary parts of the BT protocol, namely the tracker protocol, and the Mainline DHT protocol. However, VIOLA does not directly communicate with BT clients and, thus, can not take part in any file sharing activities. On the right side of VIOLA there are the torrent portal and the storage. BT portals are Web sites based on Hyper Text Markup Language (HTML) and can be directly parsed to gather torrents and metadata. Some portals, like Kickass Torrents [30], offer Rich Site Summary (RSS) feeds which can be parsed more efficiently and more reliably. VIOLA's storage is flexible as it can be readily extended. Currently storage adapters for MySQL [50] and the Apache Avro [61] file format are available.

Figure 3.10 shows the high level system architecture. VIOLA is separated into two entities: master and slave. The master is responsible for collecting new torrents,

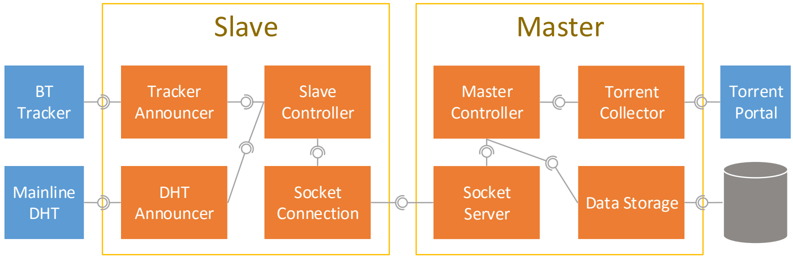


Figure 3.10: Architectural overview.

storing announce results, and orchestrating slaves. While the slave is responsible for executing the announces to trackers and the Mainline DHT and returning those results to its master. With the separation of master and slave, the horizontal scalability requirement can be addressed, as more slaves can be added if necessary. Even multiple masters can be used to monitor different torrent categories.

The master has a controller component which directs the master's behavior. The controller manages timers to collect new torrents in certain intervals and will then inform the slaves through the Socket Server component. The Socket Server also receives announce results from the slaves and returns them to the controller which forwards it to the Data Storage component, which stores it according to its configuration.

The Slave Controller is responsible for scheduling announce requests and returning them to the master through the Socket Connection component. The slave has two types of announcers, one for the Mainline DHT and the other for regular trackers. If a new type of peer source, *e.g.*, Peer Exchange (PEX), needs to be supported, additional announcers can be added.

The VIOLA architecture provides the flexibility required to support a variety of different measurements. The separation of time-consuming requests to trackers and DHT from the pure storage allows one master to control many slaves. In case the desired measurement is overtaxing a single master it can be separated by dividing responsibilities between many masters. Furthermore, the components used for storage and announcing can be easily exchanged if a different storage system shall be employed.

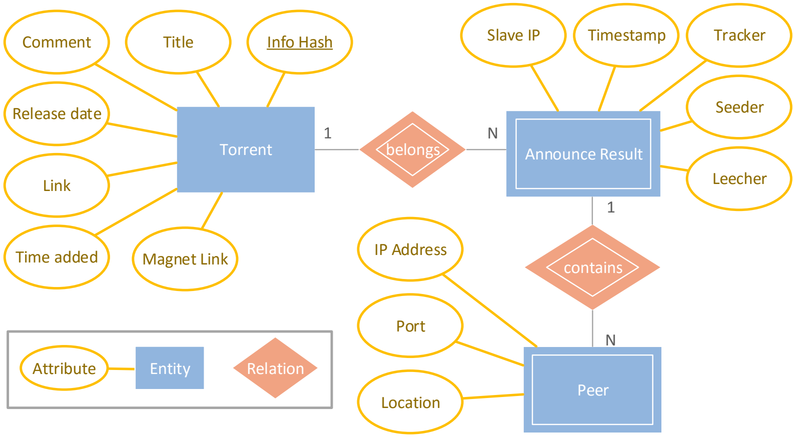


Figure 3.11: Entity Relationship Diagram using Chen's notation.

3.3.3 Data Model

The generic data model of data collected by VIOLA is described herein. This data model is oriented towards the measurement and aims to retain the complete measurement data. This generic model needs to be implemented in a specific technology, *e.g.,* MySQL, which will be discussed in Section 3.4.1 since it is implementation specific.

The data that can be collected from the BT network depends on the type of measurement. As VIOLA is a macroscopic measurement system, announce results from trackers and torrent meta information are collected. Based on those measurements the data model for VIOLA is more compact than a complete BT model such as [42]. Figure 3.11 shows the relevant entities and their relation in an Entity Relationship Diagram. The main entity is the Torrent which has many attributes, like title and release date, it is identified by the info hash which is unique for each Torrent. An Announce Result is a unique combination of torrent, slave, tracker or DHT, and time plus the contained IP addresses. Tracker results typically contain the seeder and leecher numbers of the swarm. Finally, one Announce Result contains 0 or many Peers, which are identified by IP address and port number. Additionally, a Peer can be augmented with location data resolved from its IP address.

The TORRENT entities come from torrent portals from which also the metadata can be acquired. The combination of time added and release date allows to deduce how much time elapsed between the release of the torrent and the start of its measurement. It is critical to retain the whole ANNOUNCE RESULT and not just the timestamped PEERS as information would be lost. The findings in Section 3.1 showed that it is important to know which slave received which peers to investigate the effects of the DHT. Furthermore, the tracker info regarding the swarm composition, *i.e.,* seeder and leecher ratio, is valuable information.

With the generic model presented herein, it is possible to retrace all the announces executed by VIOLA. It covers all the entities that can be measured with this measurement method. Therefore, the full swarm information collected is retained but also metadata, such as the slave who executed the announce request. Depending on the technology chosen to implement the DATA STORAGE component, the generic model needs to be customized to that technology to achieve the best possible performance.

3.4 Implementation

The key aspects of the VIOLA implementation document how the intended design was transformed into a working system. Those aspects are the applied data models in relational and row based data models, the realization of a lightweight and scalable network layer, a message protocol to enable orchestration and data flow, and the announce scheduler, which exploits the insights gained from analyzing the BTPC problem.

3.4.1 Applied Data Models

The generic data model explained in Section 3.3.3 is applied to the underlying storage technology. In the course of VIOLA's implementation, a MySQL and Apache Avro storage component have been implemented.

3.4.1.1 MySQL

MySQL is a classic relational database. The transformation of an Entity Relationship Diagram (ERD) into relational tables is straightforward. In this case the entities constitute tables just the appropriate primary and foreign keys have to be defined.

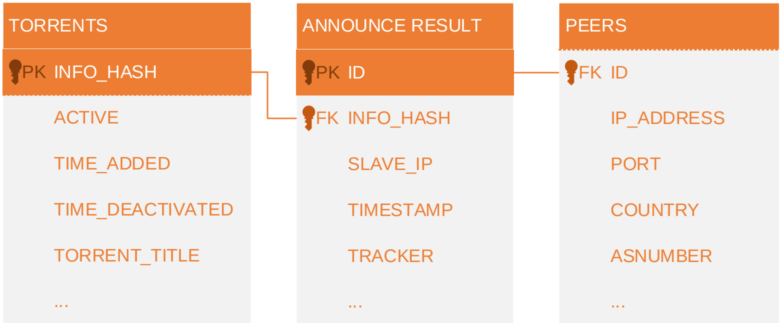


Figure 3.12: SQL table design based on the generic ERD.

Figure 3.12 shows the table design; not all columns are depicted. There is one table for each entity. The Torrent table is the main table to which the Announce Result table links by including the info hash as a foreign key. Additionally, an ID column was added to provide a unique primary key for the Announce Result table. The Peer table then uses this ID column as a foreign key, indicating the relation of a peer to an announce result.

The three tables result in frequently required join operations. Thus, proper indexing on the primary and foreign key columns is required. To increase the performance of queries, it is necessary to create indices on those columns that are used to filter the query. Typically, this would be the Info Hash, which is used to select a specific torrent, and the Timestamp, which is used to constrain the timeframe of a query. Indexing adds substantial storage overhead to the data.

The MySQL storage component is mostly suited for small measurements. First, join operations will get slower as the tables grow and the Announce Result and the Peers table will grow fast. Furthermore, the storage space required for the database and the overhead caused by indexing is substantial. The positive side of the relational model is that queries can be formulated nicely and filtering is flexible as attributes of torrents are also stored.

3.4.1.2 Apache Avro

Apache Avro is a row oriented binary storage format, supporting compression, complex data types, *e.g.,* arrays and maps, and it is very well supported in the Hadoop ecosystem. Thus, it is very well suited to store VIOLA measurements as the array of peers can be stored directly in an announce result and the join operation of the two entities becomes obsolete. Furthermore, the collected data is ready for further processing without the need of a database system.

Listing 3.1: Avro schema of an announce result.

```
{"namespace": "ch.uzh.viola.database.avro",
 "type": "record",
 "name": "ComplexAnnounce",
 "fields": [
   {"name": "timeStampt", "type": "long"},
   {"name": "announceNo", "type": "long"},
   {"name": "infohash", "type": "string"},
   {"name": "trackerURI", "type": "string"},
   {"name": "completed", "type": "boolean"},
   {"name": "seeder", "type": "int"},
   {"name": "leecher", "type": "int"},
   {"name": "total", "type": "int"},
   {"name": "slaveIp", "type": "string"},
   {"name": "slavePort", "type": "int"},
   {"name": "peers", "type":{
     "type":"array", "items":["null", {
       "type": "record",
        "name": "Peer",
        "fields": [
          {"name": "infohash", "type": "string"},
          {"name": "ip", "type": "long"},
          {"name": "port", "type": "int"},
          {"name": "asNumber", "type": ["null", "int"]},
          {"name": "continent", "type": ["null", "string"]},
          {"name": "country", "type": ["null", "string"]},
          {"name": "city", "type": ["null", "string"]}
]}]}}]}
```

Listing 3.1 shows the Avro schema definition, represented in Java Script Object Notation (JSON), of the Complex Announce, complex since it contains an array of type Peer. For the Torrent, a second schema and file are used to store torrent metadata. The Avro database implementation creates a new file every day at midnight to prevent a file from growing endlessly. VIOLA data benefits significantly from compression with a compression ratio of approximately 3, which is explained by the redundancy in the data stemming from the BTPC problem.

3.4.1.3 Best Practice

To combine the advantages of easy Big Data storage of the Avro Format with the SQL query functionality, a mixed approach is the best option. The announce results are stored in the Avro format due to their size. However, the torrent table is stored in SQLite as it allows easy querying and copying of the data. Since the number of torrents that need to be stored is in the order of 10^4, it can be handled by SQLite.

3.4.2 Socket Connection

The communication between slaves and the master is a central aspect of VIOLA as the full data has to be sent from the slaves to the master first. Thus, the master must be able to process data from many slaves, which means it must be efficient. Furthermore, the communication is required to go both ways as the master has to send updates of torrents to monitor to the slaves. Therefore, the best solution for the slave is to initiate a Transport Control Protocol (TCP), which is kept open as long as the slave is running. Having slaves initiate a constant connection allows the slaves to be behind firewalls or Network Address Translation (NAT) devices and not requiring a public IP address.

Figure 3.13 shows the class diagram of the VIOLA server socket based on the Java Non-Blocking Input and Output (NIO) [49] packages. The diagram contains only the high-level attributes and operations. The `Acceptor` is responsible for accepting incoming connections and assigning them to a `Dispatcher` from a pool. The `Acceptor` runs in a separate thread, and only one per master is required. The `Dispatcher` creates a `Connection` object for each new incoming connection and registers the `Selector` with the `SocketChannel` to capture events. Every `Dispatcher` runs in one thread; the optimal number of `Disptachers` per master depends on the load. Once an event, such as data being received, is captured, the `Dispatcher` reads the data into a buffer and executes the corresponding method in the `DispatcherEventHandler` in a worker thread. Thus, the `Dispatcher` hands off the

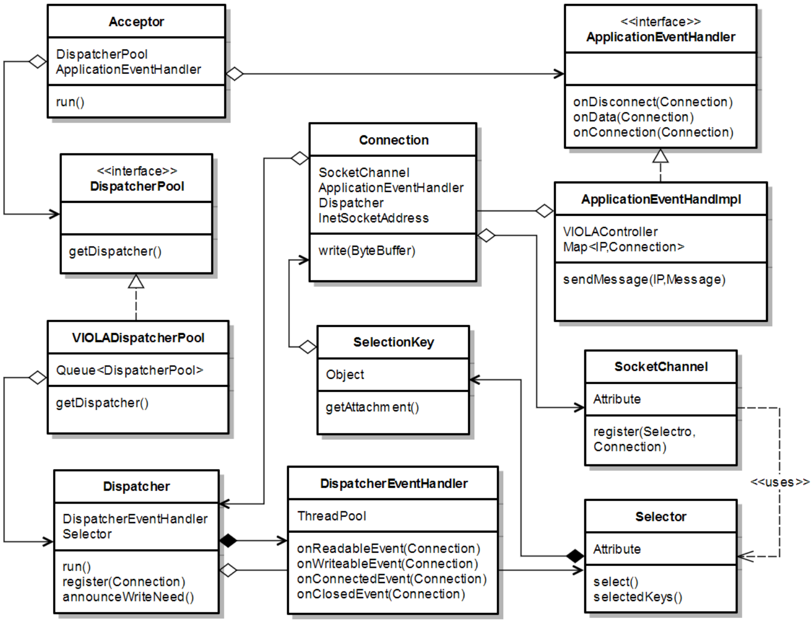


Figure 3.13: Server side socket connection class diagram.

computationally heavy operations to worker threads to stay responsive to incoming events.

Figure 3.14 illustrates the call forwarding between the different threads. The acceptor thread is only used in the beginning to handle the initialization of the TCP connection. This task is inexpensive and, thus, a single thread can handle hundreds of slaves connecting. Once the Connection is set up, incoming data is directly handled in the dispatcher thread which just reads the data from the channel and forwards a byte buffer to a worker thread which then executes the expensive operations such as message parsing and processing. This design reduces the risk of buffer overflows on the network level as the master copies incoming bytes quickly to Random Access Memory (RAM). Since the bytes are copied quickly from the limited network buffer the peaks in load occurring at the beginning of an announce interval can be handled if there is enough RAM available.

With this scalable network layer architecture, a VIOLA master will be able to process data from many slaves, provided the appropriate hardware is used. If a single

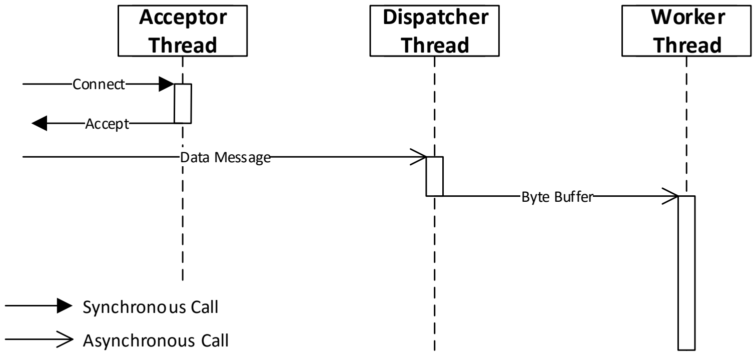


Figure 3.14: Activity diagram showing the 3 different thread levels used in the VIOLA socket server.

system is not sufficient to process the desired number of slaves, they can be split among several masters. However, in this case merging the data files later will become necessary, but this task can be executed offline and is not time critical.

3.4.3 Messages / Protocol

The communication between master and slaves consists of predefined messages of which the most important exponents shall be explained. The messages are serialized into Base64 encoded JSON before being sent to a master or slave.

Initially, a slave which opens a connection sends a register message to the master. If the slave is not yet registered and the master returns all the currently monitored torrents and their trackers in one or more schedule-torrents message,the master sends an update-slave-number message to the other slaves. The slave is now ready and will start querying trackers at the next interval. leaving slaves send an unregister message to the master so it can again send an update slave number message to the remaining slaves. In the case of an unfriendly leave of a slave, *i.e.,* a crash, the connection will timeout eventually, and the slave will be unregistered.

Listing 3.2 illustrates a typical announce reply JSON message. The message contains the metadata from the tracker reply, *i.e.,* seeder and leecher counts, plus added metadata fields like the port and IP address of the slave that returned the message but also the info hash of the torrent and the tracker URL of the tracker that provided the data. The "PEERS" field contains a list of peer IP addresses and port numbers. These IP addresses are encoded as a signed 4-byte integer since this list can contain more than 1,000 entries in case of a DHT response.

Listing 3.2: Example of an announce reply message in JSON format.

```
{ "INFO_HASH": "12069d563a9d30c1db9406a...",
  "INTERVAL_NUMBER": 20,
  "TRACKER_COMPLETE": 100,
  "TRACKER_INCOMPLETE": 200,
  "TRACKER_URL": "http://tracker1.example.com",
  "ANNOUNCE_COMPLETED": true,
  "TYPE": "ANNOUNCE_REPLY",
  "TRACKER_TOTAL": 0,
  "MESSAGE": "OK",
  "PORT": 6666,
  "PEERS": {
    "-1062731518": 6789,
    "-1062731517": 6892,
    "-1062731516": 6901,
    "-1062731515": 7001
  },
  "TIMESTAMP": 1470832700937,
  "HOST": "192.168.1.1"
}
```

Further messages are implemented to gather statistics from slaves or to remove torrents from the monitoring list if necessary. This type of message is typically used after the master executes its housekeeping routine and found swarms with sizes below the threshold. For this purpose, the master first asks the slave for a Bloom filter containing all the torrents it is currently monitoring. Thus, the master can decide which torrents need to be removed from the slave's list.

3.4.4 Announce Scheduling

The announce scheduler is responsible for deciding when to send announce requests to trackers and the DHT. There is a queue of items, torrents in this case, which need to be processed, *i.e.,* the peers of a swarm need to be discovered. This queue potenti-

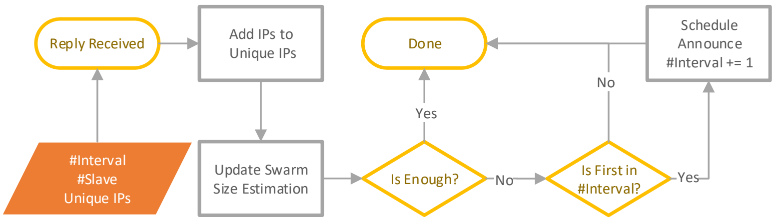


Figure 3.15: Adaptive scheduler decision taking algorithm.

ally contains tens of thousands of items, depending on the measurement setup used. A scheduler for the VIOLA system needs to work in a distributed manner, leaving two options for its deployment: (a) on the master and (b) on the slaves. The advantage of option (a) is that the master can have a complete overview of the collected peers. However, option (a) introduces control and communication overhead. Option (b) has the advantage that each slave has its scheduler, and no additional control messages are required. Therefore, option (b) is implemented in VIOLA.

The simplest scheduling algorithm executes one query for each item in its queue and then starts again. Thus, the total number of queries issued is defined by the number of trackers in the meta-info file and the number of slaves active. This approach is very simple to implement and was implemented for VIOLA instead of restarting the query process immediately. However, this simple approach suffers from the limitation that every swarm, regardless of its size, is treated the same. Therefore, either large swarms are collected only partially or there is a large overhead of over-querying small swarms. Thus, an adaptive scheduler is required that adjusts a number of queries to the swarm size of a torrent.

As shown in Section 3.1.1, due to random query responses, it is not enough to just divide the swarm size by the number of slaves to compute how many unique peers a slave has to find. Therefore, taking a distributed decision when to stop collecting a swarm is done by applying the Formula 3.6. It allows calculating the number of peers that have been collected (on average) among all the slaves, based on the percentage of peers collected from a swarm and the number of slaves active. Thus, the scheduler needs to keep track of the swarm size, the number of unique peers collected, and the number of slaves active.

Figure 3.15 depicts the process the scheduler executes when an announce result is received from a tracker. The required data inputs are the number of the interval (#interval), the number of active slaves (#slaves) and the unique IP addresses. First, the received IP addresses are added to the list of unique peers. This list is implemented as a Bloom filter, reducing the required memory since only the total number of collected unique peers and the number of newly collected peers in the response is required. With those numbers, the simple estimation technique is used to update the swarm size estimation if none is provided in the response. The simple estimation is good enough even with a few responses as it tends to overestimate swarm size. With the updated swarm size and the total number of collected peers, the slave can decide if it has collected enough to cover the swarm in collaboration with the other slaves to a specific threshold. The method that decides if enough peers were collected is presented in Listing 3.3. If more peers have to be collected, the scheduler checks if the response came from a new interval and if so increases the interval counter and initiates another round of queries to all trackers and the DHT. The interval counting is required because the queries are sent asynchronously and only the first query of one round must trigger the next round of queries.

Listing 3.3: Minimal implementation of the "is-enough" method of the adaptive torrent scheduler.

```
public boolean isEnough(int swarmsize,
    int slaves, int peers) {
   double coverage = peers/swarmsize;
    double y = 1 - Math.pow(1-coverage, slaves);
    return y >= threshold;
}
```

The number of active slaves is a more or less static parameter, and the master knows how many slaves are connected at any given time. Therefore, the master will send messages to its slaves whenever the number of active slaves changes. This approach minimizes communication overhead with the master.

With this adaptive announce scheduler it is possible to cover large swarms without the overhead of over querying small swarms. The scheduler works with swarm sizes reported by trackers and with simple estimation in case this information is missing from the tracker or only DHT announcing is used. Furthermore, the memory overhead to store state information is reduced by using Bloom filters, introducing a negligible amount of error. Finally, the threshold parameter allows adapting the scheduler to measurement scenarios in which it is sufficient to collect only a small sample of a swarm.

3.5 Discussion

With the analysis of the BTPC problem, a solution for the distributed orchestration of swarm collection was found. With a time series of tracker responses, the swarm size can be estimated accurately through a simple estimate or maximum likelihood estimation. However, since the MDHT never stores the complete swarm, the total swarm size has to be estimated with linear regression after the complete swarm has been discovered. Since this method is not feasible in distributed swarm collection, tracker responses, which are usually available, can be used, adding a margin to account for trackers having an incomplete view of a swarm.

Based on those insights, an adaptive BT measurement system, termed VIOLA, was designed and implemented. The system is capable of collecting tens of thousands of swarms concurrently (*cf.* Chapter 2). The division of responsibilities between master and slave allows deploying many relatively light weight slaves while keeping orchestration overhead small and thus scaling horizontally. This architecture also increases the resilience to hardware failure of the slaves. In case a master cannot handle the load anymore, multiple masters can run in parallel by splitting the different torrents among them, *e.g.,* according to categories.

With data collection, 'the sooner the better' is always the best answer.

Marissa Mayer

4 VIOLA Measurements and Data Quality

The Video Consumption in Overlay Networks (VIOLA) measurement system, introduced in Chapter 3, is designed to fill the gap in BitTorrent (BT) measurements identified in Chapter 2. This Chapter describes two measurements, conducted using VIOLA in 2015 and 2016, and their results summarized in Table 4.3. The 2015 measurement is based on the first version of VIOLA using a MySQL database (*cf.* Section 3.3.3) and the Simple Scheduler (SSC). With the resulting data, the performance of the measurement system, seeder-leecher ratios, and global peer distribution, are analyzed. For the 2016 measurement, the experiences gained with the former measurement were used to improve VIOLA. Those improvements are the Adaptive Scheduler (ASC) (*cf.* Section 3.4.4) and the Avro storage backend (*cf.* Section 3.4.1.2). With those results, the performance of the measurements was assessed. Furthermore, a long-term popularity and active users are investigated. Finally, the chapter is concluded, and the 2016 VIOLA measurement is added to the gap analysis chart (*cf.* Section 2.5) to show how the gap in measurements is closed.

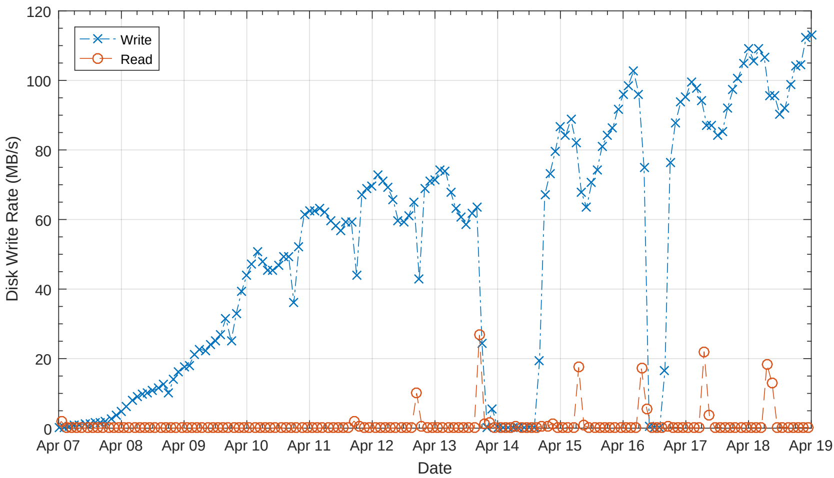


Figure 4.1: Disk writes on the database server.

4.1 2015: Simple Scheduler Measurement

Following work published in [39], the VIOLA 2015 measurements are detailed. Table 4.1 shows the main parameters of the measurement executed. The measurement period started at 19:00 hours on April 7, 2015, and lasted until 11:00 hours on April 20, 2015. The number of VIOLA slaves used was 10, which were all located at the premises of the University of Zurich. The announce interval — the time in which each slave queries trackers of each torrent — was 20 minutes. New torrents were discovered from the "Kickass Torrents" portal [30], and only content released after the start of the measurement was considered.

Table 4.1: Measurement parameters.

Start	07.04.2015 19:00
End	20.04.2015 11:00
Scheduler	Simple (SSC)
Slaves	10
Interval	20 min
Portal	kickasstorrents.to
Category	Movies

Given these settings, Figure 4.1 shows the disk read and write rate over the measurement period on the database server. From the beginning of the period, the write rate starts to rise due to additional torrents being discovered and measured. This trend continues until the end, with some exceptions. The drops in the write rate can be attributed to the data aggregation job running daily at 10:00 hours, which queries the database and locks the relevant tables. Thus, nothing is written until the maintenance query has completed. After April 12, 2015, these drops coincide with the spikes in the read rate. Due to newly added torrents, the daily amount of data written into the database is growing. Thus, the maintenance query causes more read operations and takes longer with every day. On April 14, 2015, around 19:00 hours the maintenance query took too long for the incoming write queries to be kept in memory, leading to an out of memory exception in the database system. This exception caused a crash of the master which was only discovered and fixed the next day. On April 17, 2015, the same problem appeared again. While the cause of those crashes was analyzed and corrected, the gap in the dataset during these two outages remains. The torrent discovery was not affected. During the six days before the first outage, a total of 7,977,535 unique IP addresses were observed.

Table 4.2: 2015 measurement hardware.

Resource	Master	Slave
CPU	2×AMD 6180 SE	i7-3770
RAM	64 GB	8 GB
NIC	1 Gbit/s	1 Gbit/s
IP	Public & Private	Public
Subnet	192.41.136.192/26	130.60.157.0/24
Firewall	✓	✓

The conclusion from this problem analysis is that relational databases are not ideal for the VIOLA use case. Relational databases employ sophisticated locking and protection mechanisms, assuring data integrity but introducing computational overhead. VIOLA is primarily writing static data, meaning it will not be changed once it is written. Daily maintenance and aggregation tasks read data from the last day or even older. Therefore, a row based storage format, like Apache Avro, is much better suited as files can be closed after a regular interval of time, *i.e.,* one day. Largely for those reasons, the Avro backend (*cf.* Section 3.4.1.2) was introduced to VIOLA.

Table 4.3: 2015 and 2016 measurement results overview.

Parameter	2015	2016
Duration	14 days	92 days
Samples per hour	3	3
Unique IPs	7,977,535	117,506,460
Unique Ports	65,536	65,535
Unique IP & Port	32,258,489	1,067,689,441
# Swarms	5,068	70,291
Raw Data Size	48 GB	4,7 TB
Compression	Zip	Snappy

4.1.1 Measurement Performance

The measurement produced a dataset of 48 GB size despite problems with the database system. Table 4.3 gives an overview of the data collected. In the 14 days of the measurement, approximately 8 Mio. unique IP addresses were seen in 5,068 swarms.

The analysis of the BitTorrent Peer Collector (BTPC) problem has shown that it is not trivial to collect all addresses of a large swarm. The SSC is oblivious to swarm sizes and sends 1 query to each tracker available. To find out how the SSC performed, the unique IP addresses collected in three intervals, *i.e.,* one hour, are aggregated and compared to the maximum swarm size reported by trackers.

Figure 4.2 shows a comparison between the number of peers reported from trackers and the number of unique IP addresses discovered by VIOLA for two torrents. The swarm size, as measured by VIOLA or reported by trackers, is aggregated to hourly intervals. The first torrent is "Fast and Furious 7 2015 HD-TS XVID AC3 HQ Hive-CM8" (fast7), which was the largest torrent measured with an average of 34,684 and a maximum of 45,337 peers as reported by trackers. The second torrent shown is "The Voices 2014 TRUEFRENCH BRRip Xvid-BLUB avi" (voices) with an average of 8150 and a maximum of 14808 peers reported by trackers. For voices, VIOLA discovered almost the complete swarm except for the peak in the beginning. At some points VIOLA reported even more IP addresses (08:00 hours on April 8, 2015, and 03:00 hours on April 13, 2015). This is explained by tracker failures and the fact that some peers only use DHT trackers that cannot collect swarm statistics. For fast7, VIOLA can only cover about 1/3 of the swarm due to its large size.

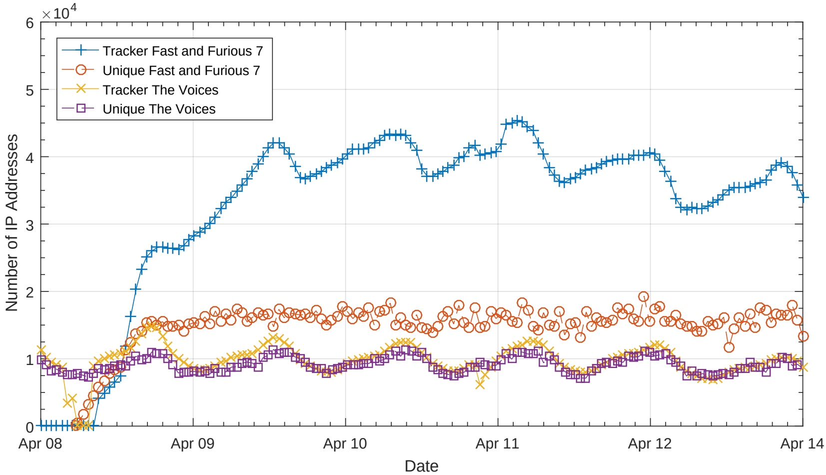


Figure 4.2: Comparison of VIOLA discovered IP addresses versus Peers Reported by Trackers.

Considering that voices is the 16th largest out of 5,068 torrents monitored in that period, this observation reveals that more than 99% of those torrents measured were accurately accounted for by VIOLA in the time frame of 1 hour. By adding more slaves or decreasing the interval, more coverage can be achieved. However, this would mean that 99% of the swarms are over queried, being inefficient and unnecessarily increasing the data size. Furthermore, a one hour time frame is long considering the effects of churn (*cf.* Section 3.1). Therefore, a more sophisticated announce scheduler, *i.e.,* ASC, is required, being able to collect swarms completely in one interval.

Although VIOLA with the SSC did not collect all swarms completely, the available data represents at least a representative sample of all swarms. Thus, the data can be used to investigate high-level BT user behavior.

To provide an overview of the number of swarms measured, Figure 4.3 depicts these figures for all swarms, swarms larger than 50, and swarms larger than 100 peers. Only the part before the database outage is shown. In the beginning, the difference between the three variants is small but gets larger over time when certain swarms start to die out. All three numbers stagnate towards the end of the last day, hinting at an equilibrium between swarms dying and new swarms appearing. However, with this short evaluation period, it is not possible to draw general conclusions. Thus, this issue will be revisited with the 2016 dataset (*cf.* Section 4.2.2).

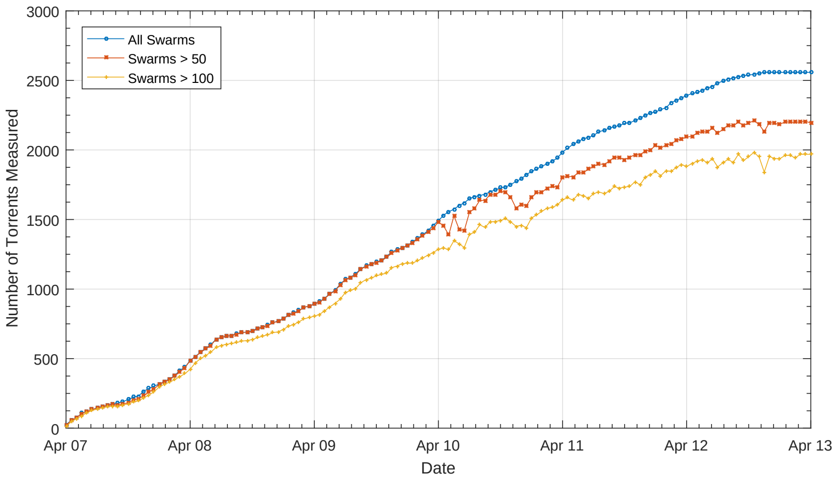


Figure 4.3: Number of torrents measured in the first 7 days of measurement in April 2015.

4.1.2 Data Analysis

For a high-level overview of the measurement period, the top 9 largest swarms covered by these measurements are displayed in Figure 4.5 according to their size over time as reported by trackers. Note that Figure 4.5 is split into two graphs, where for Figure 4.5a the y-axis' scale is twice the one of Figure 4.5b. The two gaps in the data, on April 14, and 17 2015, were caused by system failures (cf. Section 4.1). Besides the top 9 largest torrents shared during that period, the plots also shows the evolution of those torrents' swarm sizes. All swarms exhibit fast growth at the beginning of their lifetime, as seen in flash crowds. However, the swarm size does not shrink rapidly over time as flash crowds do. Swarms follow a regular diurnal pattern. Interestingly, "Fast and Furious 7 HD" shrinks to its minimum on April 15, 2015, and reaches another peak on April 19, 2015. This shows that swarm sizes, *i.e.* content popularity, does not necessarily shrink with the age of the swarm.

Figure 4.4 provides a detailed insight into the composition of the largest swarm fast7, as reported by trackers. A swarm consists of seeders — peers that have the complete file — and leechers — peers that are still downloading the file. It took three days after the release of the torrent until the number of seeders and leechers broke even. The number of seeders is constantly increasing, while the number of leechers decreases after the initial peak. Thus leechers become seeders and do not

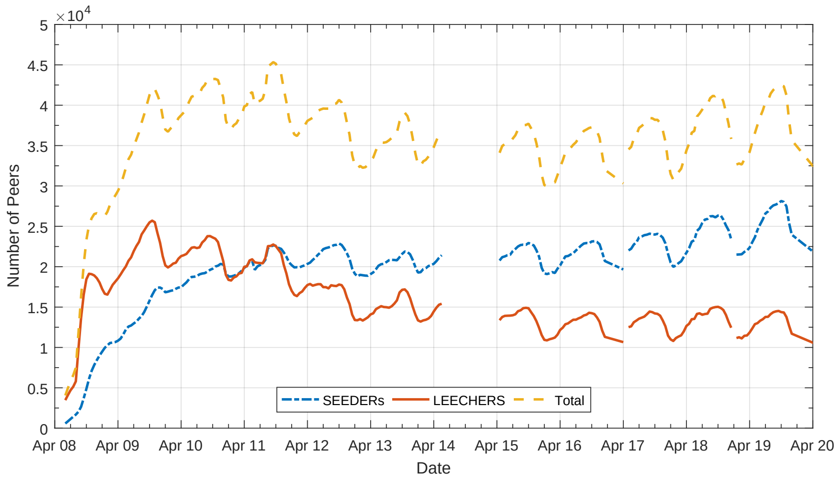
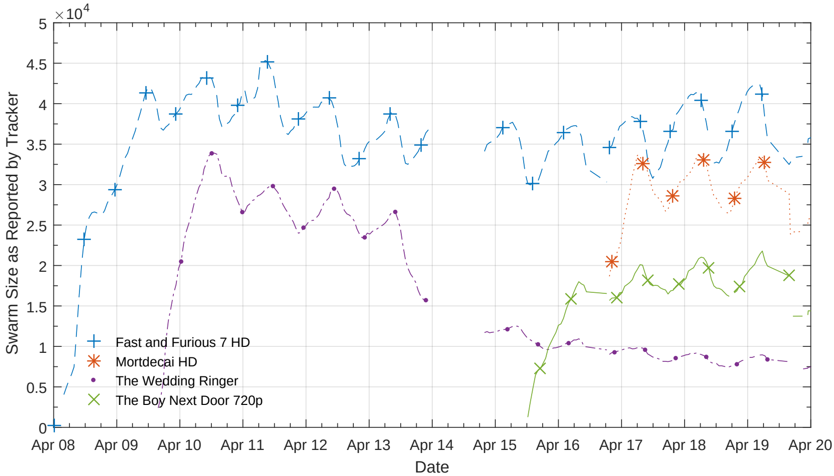


Figure 4.4: Swarm composition of "Fast and Furios 7", the largest swarm measured.

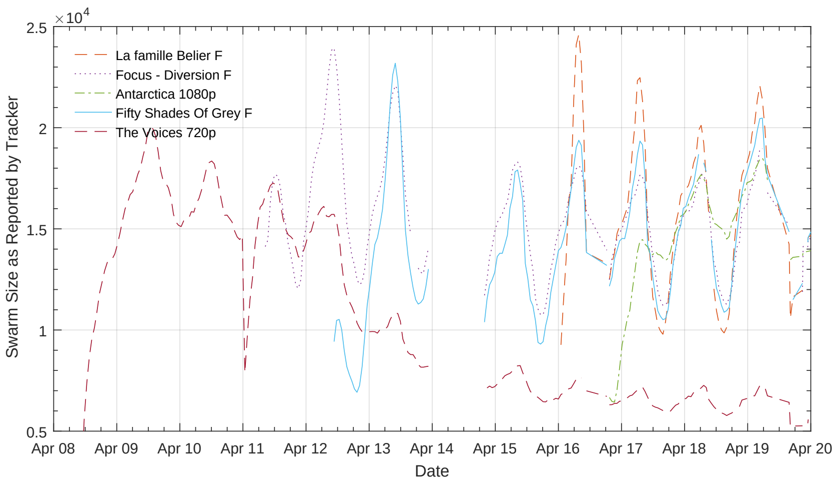
immediately leave the system after they completed their download, which can be seen in other movie or TV show torrents (*cf.* Appendix A). Peers show an altruistic behavior and free riding is not a problem in this case. Furthermore, the total number of peers increases again after April 15, 2015, which appears to be a special effect of this particular movie.

Figure 4.5b shows, among others, three French movies — marked with an F in the legend —, showing a very similar pattern, although, with different amplitudes. This observation shows the strong locality of these swarms. Since the time-wise behavior is almost equal, they must be downloaded in the same time zone. This explanation is supported by the fact that these movies are in the French language — indicated in the description of the torrent — and, therefore, mainly downloaded by users located in France. A look at the choropleth map produced by GeoChart.js (*cf.* Appendix C), shown in Figure 4.6 with data from April 13, 2015, confirms these assumptions. The ranking proves that more than 75% of all peers were located in France, Belgium, and Algeria, which all share a common time zone. These results indicate that there is a clear relation between the pattern of swarm size over time and the locality of a swarm.

The maximal swarm size of "Fifty Shades of Grey F" from Figure 4.5a (approximately 28,000) and the number of unique IP addresses shown in a choropleth map



(a) Rank 1 to 4.



(b) Rank 5 to 9.

Figure 4.5: The 9 largest swarms ranked after average swarm size.

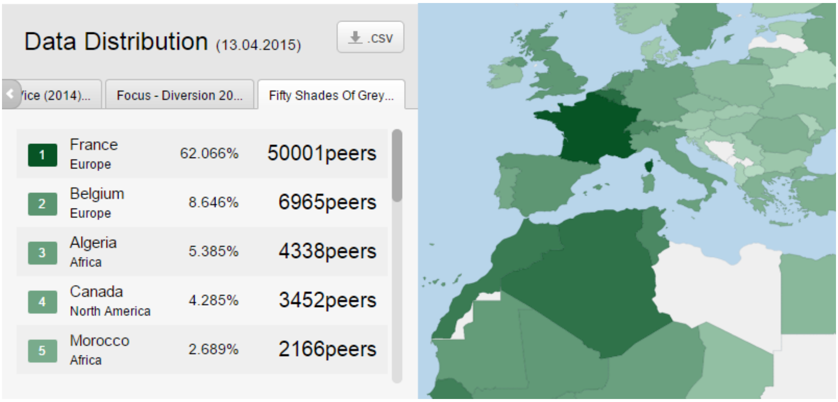


Figure 4.6: Unique peer distribution on April 13, 2015 with cubic root for "Fifty Shades of Grey F".

in Figure 4.6 for the same swarm (approximately 80,000) show a significant discrepancy. This difference is explained by the different meaning of the values. A tracker calculates the swarm size at a specific point in time, while GeoChart.js shows all unique IP addresses measured during the whole day. Therefore, there must be a high churn during a day, which leads to a swarm being completely replaced almost three times a day.

Figure 4.7 depicts the number of unique IP addresses measured per continent over the 24 hours of April 13, 2015. Although India had the most unique IP Addresses on that day, Europe in total had more than Asia. The time zone patterns are clearly visible, even for those continents with few IP addresses, *e.g.*, Oceania (OC) and South America (SA). North America (NA) and SA are very much in sync with their peak at 04:00 hours, followed by Asia (AS) and Europe (EU). Europe, spanning 3 hours in time difference, has the narrowest peak while Asia, spanning 9 hours, has a very smooth peak. NA and the other continents with even fewer peers show smooth transitions as well.

Comparable measurements were undertaken with the Ono plugin [51] in November 2008 and 2010. The dataset used in [51] was collected from the viewpoint of individual peers and does not allow for the identification of the content shared. However, the continental diurnal patterns can be compared and show similar behavior. A notable difference is the order of continents concerning the number of peers they contribute. While Europe was already the largest contributor, AS has overtaken

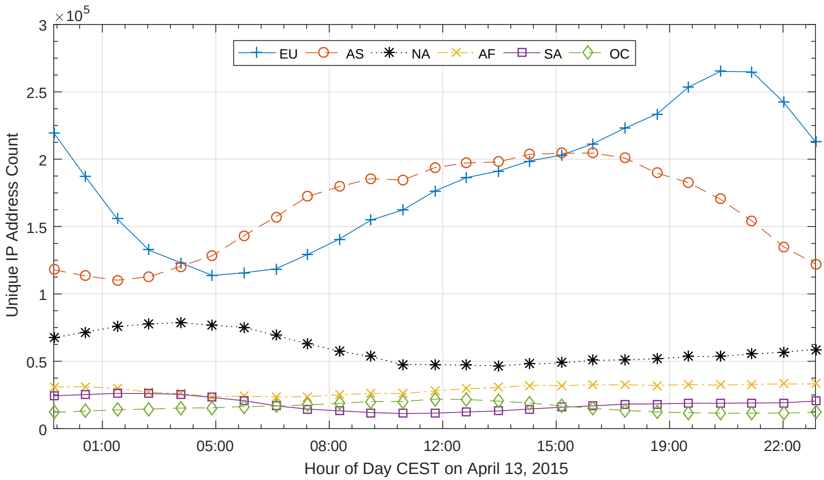


Figure 4.7: Diurnal pattern of unique peers per continent on April 13, 2015.

NA. This trend was already visible from 2008 to 2010, were NA and AS were on the same level. The growth of AS compared to NA is influenced by several factors: file sharing prosecution in NA, emerging video streaming services in NA, or improved internet connectivity in AS. Those differences between the Ono and VIOLA measurement show that it is necessary to measure BT consistently over years, or at least repeat measurements regularly, to identify long-term usage trends.

4.2 2016: Adaptive Scheduler Measurement

The Adaptive Scheduler (*cf.* Section 3.4.4) implements the BTPC solution (*cf.* Equation 3.6) which allows to efficiently collect large and small swarms simultaneously. A VIOLA measurement using the Adaptive Scheduler was conducted from May 1, 2016, until July 31, 2016. The measurement parameters are outlined in Table 4.4. Compared to the 2015 hardware, the slaves have a similar amount of resources at their disposal. The most critical resource is RAM since the Adaptive Scheduler needs to keep more state information in memory than the simple scheduler.

Additionally, a scraper for The Pirate Bay was added, increasing the number of torrents that can be discovered and improving reliability in the case of a portal being shut down. However, many torrents discovered in one portal were also found in the other. The main differences to the 2015 measurement are:

Table 4.4: 2016 measurement parameters.

Start	01.05.2016 00:00
End	31.07.2016 23:59
Slaves	10
Interval	20 min
Portal	kickasstorrents.to, The Piratebay
Trackers	UDP, DHT
Category	Videos
Threshold	95%

- Use of the Adaptive Scheduler with 95% threshold
- Use of Kickass Torrents and Piratebay
- Duration of 3 months

A minor difference to the 2015 measurement is the execution environment of the slaves. For the 2016 measurement, the slaves and the master were connected to the same private subnet, making the communication more reliable and secure. With this move, the slaves were changed from physical to virtual machines, which has no influence on the measurement as long as the provided resources are enough. The hardware used for the 2016 measurement is detailed in Table 4.5.

Table 4.5: 2016 measurement hardware.

Resource	Master	Slave VM	VM Host
CPU	2×AMD 6180 SE	Virtual 4 cores	2×Intel Xeon E5520
RAM	64 GB	8 GB	48 GB
NIC	1 Gbit/s	Virtual	2×1 Gbit/s
IP	Private	Public & Private	Private
Subnet	—	192.41.136.192/26	—
Firewall	✓	×	✓

4.2.1 Measurement Results

The three-month measurement produced a dataset of approximately 4,7 TB size. The key metrics of the dataset are given in Table 4.3. In those three months approximately 16 million unique IP addresses were recorded, but over 1 billion unique peer

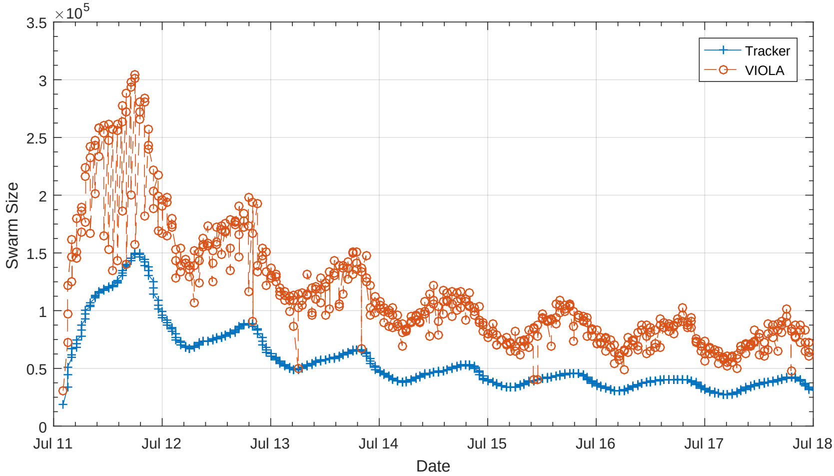


Figure 4.8: The first 7 days of the largest measured swarms lifetime show that peer addresses collected by VIOLA are higher than the swarm size Reported by Trackers.

addresses – consisting of Internet Protocol (IP) address and port – were collected. A total of over 70,000 torrents were discovered.

With the ASC even the biggest swarms are expected to be collected to 95%. The biggest swarm, as reported by trackers, was released on July 11, 2016, shortly after midnight UTC and reaching a maximum swarm size of 150,000 according to trackers and more than 300,000 according to VIOLA. This swarm size is more than six times larger than the biggest swarm (fast7) from the 2015 measurement. The content shared in this swarm is Game of Thrones (GoT) episode 8 of season 6 in High Definition (HD) quality, *i.e.,* "Game.of.Thrones. S06E08.HDTV.x264-KILLERS[ettv]" (GoT S06E08). Analogous to Figure 4.2, the swarm size reported by trackers is compared to the unique peers collected in each interval. Peers are identified by IP address and port number, *i.e.,* their peer address.

Figure 4.8 shows the maximum swarm size reported by trackers, *i.e.,* Tracker GoT S06E08, and the unique IP and port combinations measured by VIOLA, *i.e.,* VIOLA GoT S06E08, for each interval of 20 m. In contrast to the SSC, the ASC collected more peers in a single interval than were reported to be in the swarm by trackers, with a few exceptions where the numbers are equal. Collecting more peers than advertised is possible due peers using different sets of trackers and some peers exclusively using Distributed Hash Table (DHT) for peer discovery. Considering

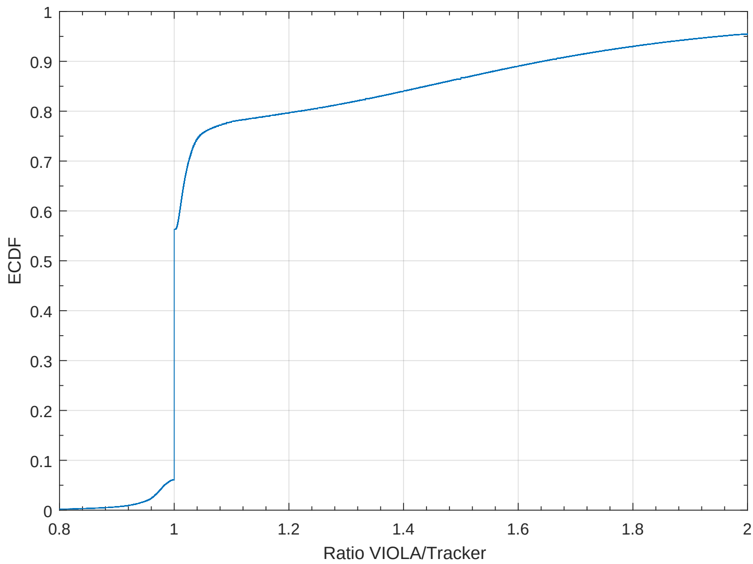


Figure 4.9: Fraction of swarm discovered by VIOLA in 20 minutes intervals.

that this result was achieved with the same amount of slaves, it can be concluded that the ASC greatly improves measurement performance over theSSC and even the largest swarms can be fully collected. However, the GoT S06E08 swarm is only one example, and the overall performance across all swarms needs to be investigated.

To investigate the overall performance, the collected peers are compared to the maximal swarm size reported by trackers during each interval. For this analysis, the maximum swarm size reported by any tracker is considered 100%. Thus, the number of peers collected by VIOLA is divided by the swarm size resulting in the fraction of the swarm collected by VIOLA. This fraction can be greater than 1 since trackers are not always aware of all peers in a swarm. Figure 4.9 shows the Empiric Cumulative Distribution Function (ECDF) of those fractions over all intervals. The first observation is the vertical line at fraction 1. This step in the ECDF is due to 50% of all fractions being equal to one. Assuming that trackers report correctly if a swarm was not fully collected, in 56% of cases the swarm size reported by trackers is correct since VIOLA measured exactly the same. Taking a look at the lower end, *i.e.,* the swarms that were not fully collected, only 6% of swarms showed a fraction lower than 1, and only 1.8% of swarms were collected to less than 95%. However, since those samples measured below 95% are relatively few and not only occurring in large swarms, they can be considered noise rather than limitations of the measurement methodology itself. This noise partially stems from bogus tracker swarm size reports (*cf.* Section 4.2.2). That leaves 45% of data points which were collected above the reported swarm size. The reasons for this are the same as for the GoT S06E08. Finally, with a goal of collecting swarms to 95%, defined by the threshold, over 98% of data points reach this goal. This result can be considered excellent as one very accurate snapshot of the BT system can be collected within 20 minutes.

This outcome can be attributed to the Adaptive Scheduler, solving the BTPC problem more efficiently than the simple scheduler. By adapting the number of queries to the swarm size, or the estimate if none is available, resources can be directed towards the large swarms where they are most needed. Apparently, trackers do not have a complete overview of more than 40% of swarms, shown by the cases which have a ratio larger than one. Only a few samples show a ratio above two. Analysis of the relation of ratio to swarm size did not show a linear correlation between the two factors. Thus, the large ratios can only be explained by different trackers or none at all being used by different parts of a swarm.

4.2.2 Data Analysis

Popularity of content in Peer-to-Peer (P2P) systems is generally interpreted as number of downloads [15]. However, it is impractical to identify the total number of downloads in BT as it is hard to reliably identify single users over days due to changing peer addresses. Identifying a user reliably during a 20-minute interval can be done by peer addresses since the IP address or the port number typically only change when the client is restarted, or the internet connection is reconnected. Therefore, for the investigation of content popularity, the number of concurrent users in a 20-minute interval is used as popularity metric. Furthermore, for each swarm, the maximum size or the maximum of unique peer addresses encountered in the investigated time frame is used.

Figure 4.10 shows the rank popularity of all torrents encountered in the measurement on a log-log scale. A distinction is made between the swarm sizes reported by trackers and the maximum unique IP and port pairs. Confirming the results of Section 4.2.1, the VIOLA measurement results show consistently higher maximum swarm sizes than the tracker responses. The VIOLA rank popularity shows a more

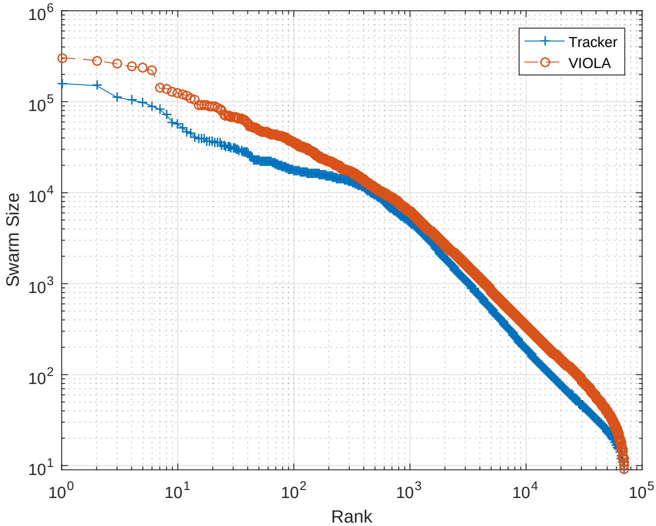
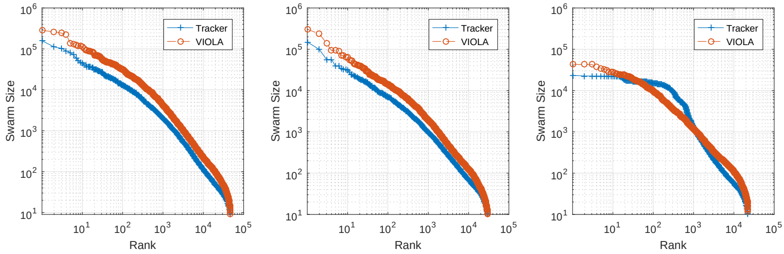


Figure 4.10: Ranked swarm sizes of all measured torrents between May 1, 2016 and July 31, 2016.

consistent shape than the rank popularity for trackers. In accordance to [15], both curves do clearly not fit a power law distribution, which would constitute a straight line. However, there are few very popular torrents and many unpopular torrents. This is typical for a content ranking and has been observed in BT [15] and Youtube videos [7] before. The pareto-principle comes to mind that was observed in previous work [24]. The cut-off long-tail is partially explained by the removal of swarms with less than 30 peers.

To illustrate popularity rank distribution on a different time scale, Figure 4.11 presents the rank popularity for May, June, and July 2016. Figures 4.11a and 4.11b show a very similar distribution to the full three month distribution. However, the July distribution shows differences to the other two. The July VIOLA popularity still has a similar distribution to the previous months while being lower in general. The tracker reported popularity distribution for July shows a different shape, showing higher popularity for some torrents than VIOLA. Since this effect does not appear in the most popular torrents, a problem with tracker reports is the most likely explanation.



(a) May 2016 (b) June 2016 (c) July 2016

Figure 4.11: Torrent popularity on the three month of measurement.

Figure 4.12 presents the average swarm size for swarms larger than 1,000, which were reported by five trackers seen from the beginning until the end of July 2016. "Opentrackr" and "Copper Surfer" are the most stable trackers overall since in the first ten days those two are the only active trackers. On July 11, two new trackers became active: "Leechers-Paradise" and "Zer0Day". "Zer0Day" stays very close to "Copper Surfer" and reports only slightly higher average swarm sizes than "Copper Surfer". However, "Leechers-Paradise" starts off slightly lower than the others but reports three times higher average swarm size than "Opentrackr" and "Copper Surfer", which does not seem to be correct. After July 26, the three top trackers report average swarm sizes in a similar range. On July 20, 2016, the administrator of the Kickass Torrents (KAT) portal was arrested in Poland, and the various domains of the portal were seized successively [6, 21]. KAT was a pure portal and did not operate a tracker. Thus, it is surprising to see two new trackers emerge a few days before this incident.

The emergence of two new trackers among the top trackers could be a result of the investigations and actions against KAT, forcing users to shift to another portal using different trackers. This effect would lead to a slightly higher swarm size in the trackers from the new portal as new users are joining. However, there is no indication that one tracker suddenly lost popularity. Furthermore, the increase in swarm size on the "Leechers-Paradise" tracker between July 16 and 26 is over proportionally high. Thus, a technical issue with the tracker is the better explanation. Finally, "Leechers-Paradise" shows a slightly higher swarm size than other trackers in the period from July 26 to 31, supporting the theory of users from different communities adding a new tracker to their .torrent files.

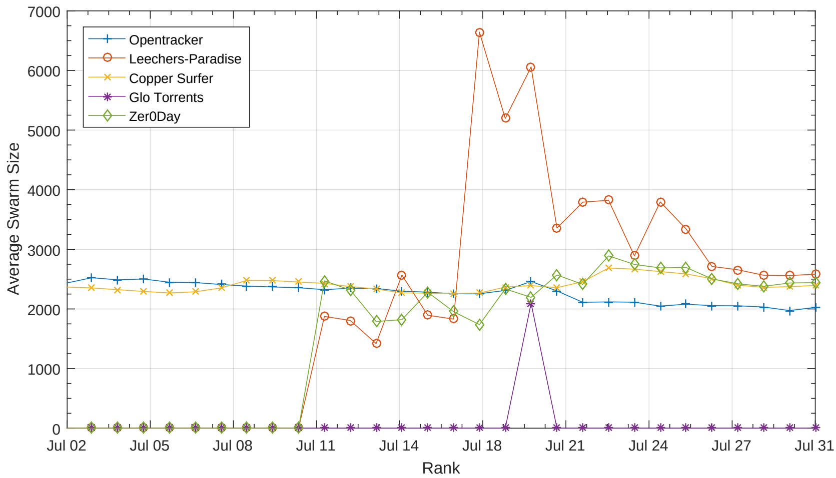


Figure 4.12: Average swarm sizes reported by trackers for swarms larger than 1,000.

Finally, the rank popularity of measured torrents and tracker reported torrents follow a distribution comparable to previous findings [7]. Thus, the VIOLA measurements confirm that popularity does not follow Zipf's law as the plots in Figures 4.10 and 4.11 are clearly not linear. Specifically, the head and the tail of the distributions are nonlinear.

4.2.3 Long Term Usage Patterns

The three-months measurement data provides a different perspective on diurnal (*cf.* Section 4.1) and longe-term patterns. As the BT environment is very dynamic, a closer look at the number of torrents collected during the three month period is required.

Figure 4.13 depicts the number of torrents that were actively measured in each hour of the whole measurement period. A distinction is made between all swarms, swarms larger than 50 peers (>50), and swarms larger than 100 peers (>100). The number of total torrents grows rapidly from the start of the measurement and keeps rising until the beginning of June when it decreases rapidly. Swarms larger than 50 peers do not show this behavior. Thus, the cleanup task not working properly until June explains this effect. This error is not a problem since it only means that small swarms are also included in the measurement. Swarms greater than 50 show a much

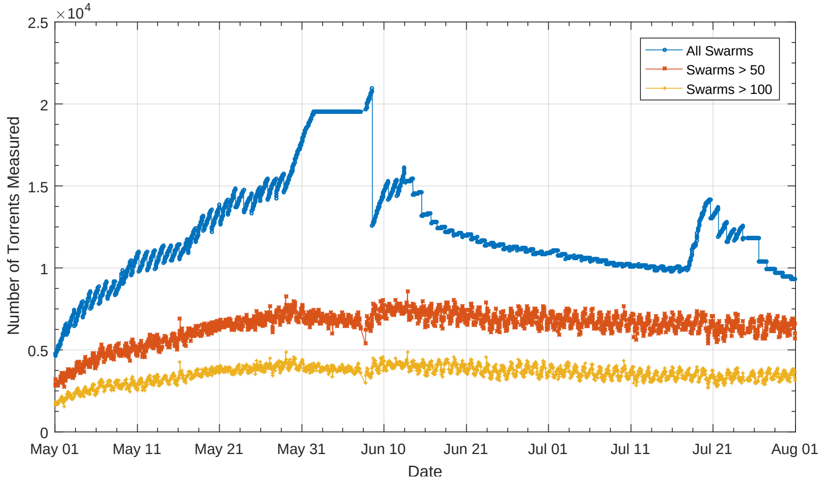


Figure 4.13: Number of torrents measured over the whole measurement period.

more stable behavior. As expected, the number of swarms greater than 50 increases at the beginning of the measurement until the end of May after which the number stays stable and even starts to decrease slightly. This decrease is most likely a seasonal effect due to summer vacation time in which fewer content is released. This investigation shows that the number of swarms larger than 50 remains stable within certain bounds. Thus, those small swarms do not have a noticeable effect on total peers in the system and also not on data traffic.

Figure 4.14 shows peers measured in a day per continent, (*cf.* Appendix B for hourly resolution). Surprisingly, Asia (AS) constantly contributes the most peers compared to the 2015 measurement (*cf.* Figure 4.7). Furthermore, Africa (AF) and South America (SA) have similar numbers compared to North America (NA). This observation is partially attributed to the broader scope of torrents that have been measured. However, the bulk of peers remains in Europe (EU) and Asia. Oceania (OC) does not play a big role. On all continents, a distinct weekly pattern can be observed, showing a peak on Monday of every week, *e.g.,* May 2, 2016. The pattern weakens towards the end of the measurement. This effect is surprising as Mondays are usually working days. However, it can be explained by the weekly release of the hit show "Game of Thrones Season 6" [26], which is released Sunday nights in the US until the season finale on June 26, 2016. This observation leads to the conclu-

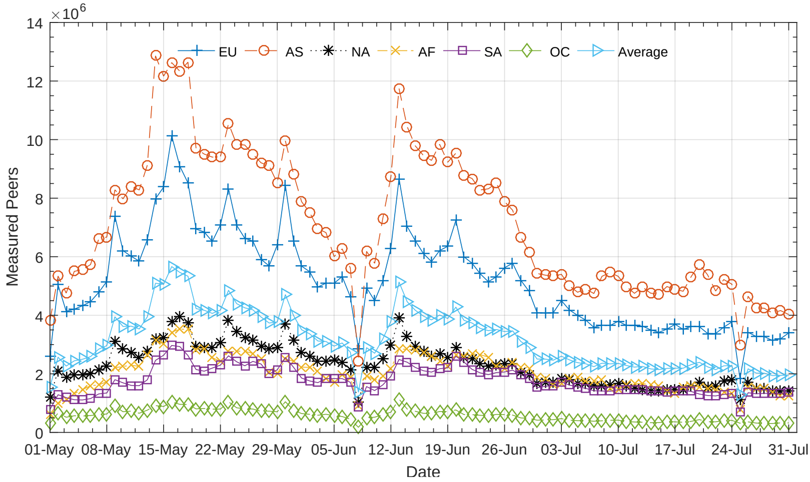


Figure 4.14: Number of peers per day and continent showing a distinct weekly pattern due to the releases of new Game of Thrones episodes.

sion that release dates of popular Television (TV) shows have a larger effect on user behavior than the day of the week. There are two very low values on June 8, 2016, and July 25, 2016, which are artifacts of system outages in the measurement infrastructure and not real observations.

4.3 Discussion

The 2015 measurements showed that even with a simple scheduler, many swarms can be collected to a high degree. However, the 2015 dataset is not comparable to the 2016 dataset since the latter is much more extensive and includes a complete set of video related torrents. This comparison becomes apparent when looking at Table 4.3 showing a difference factor in the order of tens in the unique peer addresses collected and the number of torrents discovered. However, the number of torrents monitored reached an equilibrium in both cases, at least for swarms larger than 50 (*cf.* Figure 4.3 and 4.13). Comparing Figure 4.7 and 4.14, a trend can be identified. South America and Africa have caught up with North America in the number of peers. Furthermore, Asia has become the leader in the number of active peers a day, stressing again that regular repetition of BT measurements is critical. One explanation can be found in the content shared, as Game of Thrones is available in North

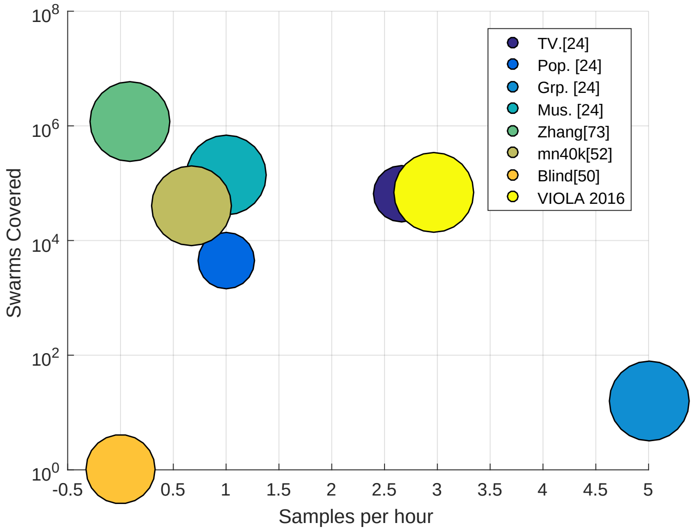


Figure 4.15: Visualization of measurement studies including the VIOLA 2016 measurement presented herein.

America through legal channels, fans of the show living in other continents have no way to watch the show other than piracy.

The implementation of the adaptive scheduler solving the BTPC problem increased the capabilities of the collector immensely. It is now possible to take snapshots of 20,000 swarms in less than 20 minutes and, thus, collect time series of those swarms. Furthermore, scraping the DHT results in more than 40% of swarm collections being larger than the largest swarm size reported by any tracker. Those trackers do not always provide correct swarm size numbers, which could have an impact on measurement studies that rely on tracker crawling. However, such false reports can be identified by comparing results from different trackers (*cf.* Figure 4.12). Furthermore, trackers often track only a fraction of a swarm and, thus, cannot report the real swarm size. Collecting the DHT shows that 45% of swarms are bigger than reported (*cf.* Figure 4.9).

The data collected over those three months comprises an original BT dataset. Except for the outages on June 8, 2016, and July 25, 2016, three detailed snapshots of an average of 12,000 swarms were taken every hour. Accordingly, Figure 4.15 presents an overview of existing BT measurements including the VIOLA 2016 measurement presented herein. The VIOLA 2016 dataset clearly closes the gap (*cf.* Section 2.5), it is very close to the TV dataset [24] but more detailed. Thus, it can be concluded that the VIOLA measurement system combines the three dimensions, time, content, and user, unlike previous measurements which allow novel analysis methods to be applied to this data. The main advantage of the VIOLA 2016 dataset is the continuous coverage over three months which allows analyzing changes over time in more detail than before. Chapter 5 details procedures and explains the methods necessary to make those changes visible.

The world is one big data problem.

Andrew McAfee

5

Content- and User-Centric Views of BitTorrent

The VIOLA dataset collected from May to July 2016 (*cf.* Chapter 4) provides very detailed swarm information, *i.e.,* three complete collections of peer addresses of thousands of swarms per hour. While initial investigations into the data were already presented, a deeper analysis of the data is provided herein to show what novel insights can be gained with the VIOLA data. For this purpose, Social Network Analysis (SNA) is applied to the VIOLA 2016 dataset to show the dynamics of the BitTorrent (BT) system.

SNA traditionally investigates individuals and their social relations, which are modeled as graphs or networks consisting of nodes (individuals) and edges (relations, *e.g.,* friendship) [57]. Modeling a system as a graph allows to apply SNA or graph measures on that system, indicating features of the graph or its nodes and their interconnections, termed network- and node-centric measures respectively. The main limitation of most SNA measures is that they can only be applied to graphs with one type of node, termed a one-mode graph [57]. Libraries to calculate these measures [66] are well supported and tested and can be easily applied to any one-mode graph. Therefore, SNA methods are not limited to social networks [70]. Furthermore, SNA measures are typically dependent on the size of a network, *i.e.,* the number of nodes it contains, and, thus, are mainly suited to quantify and compare similar networks, *i.e.,* the same network in different timeslots.

SNA methods are predestined to analyze a Peer-to-Peer (P2P) network such as BT since it consists of connections like a graph. However, before applying those methods to the VIOLA dataset, the data needs to be transformed into a one-mode graph. Two possible projections of the two-mode graph, *i.e.,* peers connected to torrents, are investigated. The first is focusing on the peers or rather the locations where content is shared, providing insights in the locality of content. The second is focusing on the content itself and, thus, indicating the popularity of content, which is the foundation for recommendation systems [16]. The transformations required to create one-mode graphs out of the Video Consumption in Overlay Networks (VIOLA) measurement data can be implemented in a three stage MapReduce job. Therefore, this method applies to the VIOLA data as well as larger datasets since the processing can be scaled horizontally.

5.1 Data Transformation Methodology

A generic method to transform tuples combining a User ID (UID), a Content ID (CID), and a timestamp is described, following the ConNet principle published in [38]. Thus, this method can be applied to the VIOLA dataset but is not limited to only this use case. Furthermore, changes to the methodology are possible if more data is available, *e.g.,* Section 5.2.2 uses the content size to give another dimension to the edge weights.

Figure 5.1 presents the BT network as a two-mode graph (in the middle) consisting of Content and Consumer nodes and edges representing a sharing relation. On the left is the consumer-centric projection, where each consumer sharing a content receives an edge to every other consumer sharing the same content. Duplicated edges are aggregated, and the number of duplicates is used as the weight of that edge. This graph can be used to discover content to recommend to a user by following the edges, similar to the YouTube recommendation system described in [16]. On the right side of Figure 5.1, the content-centric projection is shown. It follows the same procedure, the only difference is that the peers are turned into edges, and the tor-

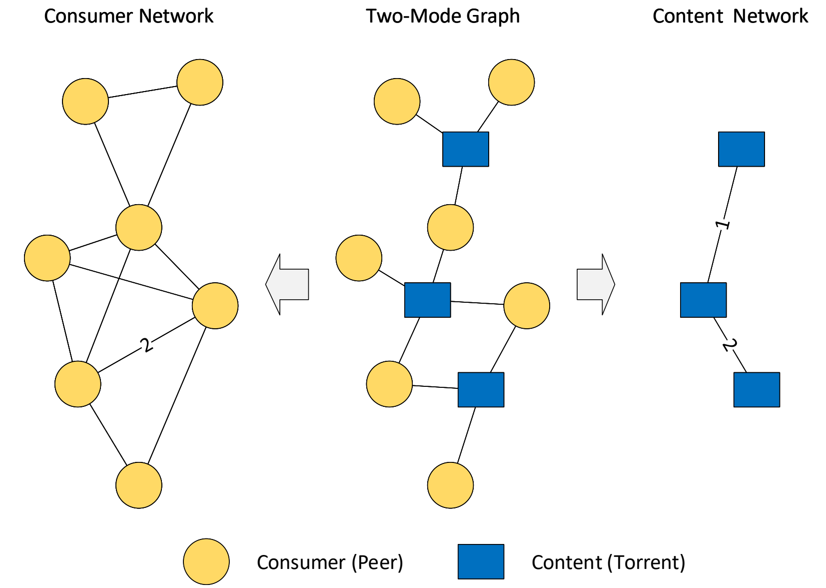


Figure 5.1: Abstract BitTorrent two-mode graph and its torrent and content network projection.

rents remain nodes. This content-centric network can be used to investigate with whom peers share most content. Since peers, or Internet Protocol (IP) addresses, can be mapped to countries or Autonomous System (AS) numbers, the network can be further abstracted to generalize the analysis.

To detail this process, the method is explained according to three MapReduce stages, allowing for a scalable implementation and execution in data processing clusters. To prove the applicability of those stages, a Scala implementation for Apache Spark is presented. Since spark can pass data in memory between the three stages, it executes faster than traditional MapReduce, although the resulting graphs are equal.

5.1.1 Transformation Design

Figure 5.2 depicts the generic transformation method. The mapping process and its stages are detailed as a process to be executed for a specific timeslot. The processing of the stages suits a MapReduce model and can, thus, scale to large data sets. The details of the individual stages are explained.

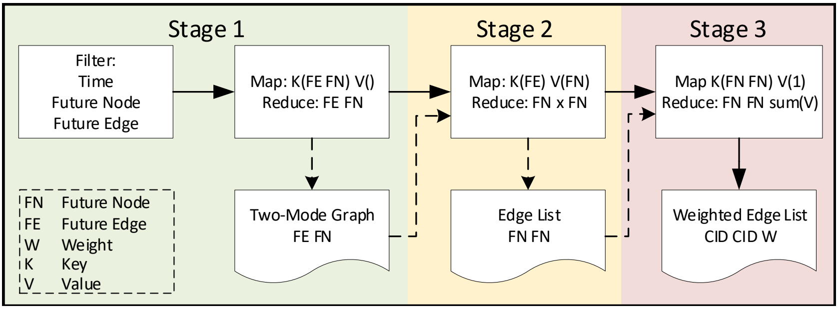


Figure 5.2: Diagram of the projection transformation flow.

Stage 1 transforms a set of tuples into a two-mode graph, depicted in Figure 5.1. Depending on the data format of the records, Stage 1 can be left out. However, in the most general case, data can contain redundancies, as in the VIOLA case, which can be elegantly removed by a MapReduce job or a Spark SQL query, handling the filtering as well. Filtering is generally necessary since certain timeslots will be investigated and data can contain noise or outliers, which need to be removed. If there is more than one timeslot, the filter can be changed and the full process can be repeated for every timeslot. Other attributes of records can be used for filtering (*cf.* Section 5.1.3), *e.g.,* a specific set of users or the location of users if it is available. The MapReduce job abstracts the content providing service users who are connected to content if they have consumed it, *i.e.,* watched the video. Therefore, the Map phase maps each record to a concatenation of CID and UID. For each combination of CID and UID, the Reduce phase emits an edge into an edge list.

Stage 2 implements the projection of the two-mode graph into a one-mode graph, as depicted in Figure 5.1. The projection removes the user nodes from the two-mode graph and replaces them with fully meshed connections between the content that users consumed. Semantically, such an edge means "users also watched" similar to the Youtube recommendation system [16]. An analogous example is the "bought together" used in online shops. For this purpose, the Map phase in Stage 2 maps UIDs to the CID. Thus, the Reduce phase receives a list of CIDs for every UID and returns an edge for each pair of CIDs found, *i.e.,* a permutation of the unique CIDs per UID. The result is an edge list containing an edge for every CID that had a common UID.

Stage 3 reduces the size of the edge list by aggregating the edge weights. Potentially, there are many parallel edges in the edge list from Stage 2. To reduce the number of edges, they are combined, and the number of parallel edges is used as weight. Stage 3 is even more important if peers are aggregated to ASes or countries. Thus, another MapReduce job is required, optimizing the size of the output. This job is similar to the word count algorithm; the value 1 is mapped to each edge, the reducer sums all the values for an edge and emits the edge and the final sum, which corresponds to the weight of that edge.

This simple edge list can be used to import the graph into a graph analysis software, *e.g.,* iGrpah [66]. This generic method can be applied to any data series consisting of CID, UID, and time tuples. Depending on data size and available tools, an implementation looks different. For the VIOLA dataset, a Spark based implementation was chosen.

5.1.2 Implementation for the Country Network

The VIOLA data is in the row based Avro format [61], which can be directly used by Hive [62] or Spark SQL. To improve the performance of the transformation, the redundancy is removed, and the resulting tables are stored in the columnar Parquet [63] storage format. As a result, the queries became faster due to a reduction of data (removed redundancy) and a reduction of data needed to be read for a single job (only relevant columns are read). The three stages were initially implemented using Hadoop [65] MapReduce jobs, which resulted in one MapReduce job for each stage of the method. The drawback of this approach was that output had to be written to the file system after each job and then read again for the next job. Thus, execution time was long and storage overhead high since the intermediate files were not relevant after job completion.

The solution to those problems is the implementation of the three stages in Apache Spark [64]. Spark is much more flexible than traditional MapReduce as it allows more types of transformations. Furthermore, several transformations, or jobs in MapReduce, can be chained together without the need of defining multiple jobs. Since transformations can be directly chained, the complete data is kept in memory, improving performance greatly compared to MapReduce. Another convenience offered by Spark is Spark SQL, which allows querying files in the HDFS or Hive tables. Thus, it is well suited to implement the filtering.

Listing 5.1: Scala implementation for Spark.

```
for(month <- 5 to 7){
 val days = 1 to maxDayof(month)
 days.foreach(day => {
   val query = "SELECT A.infohash, A.country " +
     "FROM torrentsperip as A " +
     "JOIN dailysharedtorrents as B " +
     "ON ( A.peeruid = B.peeruid " +
     "AND B.year = A.year " +
     "AND B.month = A.month " +
     "AND B.day = A.day ) " +
     "AND A.year = 2016 " +
     "AND A.month = " + month + " " +
     "AND A.day = " + day + " " +
     "AND B.shared between 1 and 50 " +
     "GROUP BY A.infohash, A.country"
   val pt = sqlContext.sql(query)
   val stage1 = pt.select(pt.col("infohash"),
        pt.col("country"))
    .where(pt.col("country").isNotNull)
    .and(pt.col("country").notEqual("null")))

   val stage2 = stage1.map(
     record => (record(0).toString, record(1).toString))
     .groupByKey().flatMap{
      case (infohash: String, countries: Iterable[String]) =>
       Perm.permutation(countries)
     }

   val stage3 = stage2.map(edge => (edge, 1))
     .reduceByKey(_ + _)
     .map(edge => edge._1.from + '\t' +
        edge._1.to + '\t' + edge._2)
   stage3.saveAsTextFile(outputpath)
 })
}
```

Listing 5.1 shows the essential part of the Sark implementation for a country based network, meaning that countries will be future nodes and torrents commonly shared among countries constitute the future edges. Stage 1 consists of a Spark SQL query defined on line 6. The query joins the "TorrentsPerIP" and the "DailySharedTorrents" tables. The former is used to filter certain peer addresses that share a number torrents above a certain threshold, in this case 50. The latter contains all unique IP addresses per torrent measured in one day. Thus, the query can run efficiently and makes filtering simple. The query result is again sub-queried in line 19, effectively filtering missing or non-existent country values. Since the actual query is executed lazily this does not affect performance. The result is a list of all unique torrent and country pairs, *i.e.,* a two-mode graph's edge list.

Stage2 first maps all countries to the torrents they share. Thus, the "flatMap" operation receives an array of countries for every torrent. Each torrent is mapped to all the combinations ("Perm.permutation(...)") of countries in the array in line 30. Those combinations are the edges of the newly projected one-mode graph.

Stage3 maps each edge to a value, corresponding to weight one. The edges are then reduced by key, meaning that all equal edges are aggregated, and the weights are added resulting in the final weight per edge. Edges and weights are written to a text file in the Hadoop Distributed File System (HDFS). The map in line 29 is necessary to produce the correct line syntax.

5.1.3 Filtering

Before using BT data for analysis of user related effects, such as their preferences or behavior, it has to be investigated how the BT network is used. Since BT is an open system, it is being used in other ways than originally intended, *e.g.,* measurements. Measurement systems querying trackers will register their IP address with the tracker and appear to be a member of the swarm. In the case of VIOLA this would be easy to filter as the slave's IP addresses are known. However, there can be other measurement systems at work, or so-called seed boxes can bias the analysis.

Typically, regular users download content they want to watch, this is expected to be the largest group of peers in the network and is also the most relevant one. Furthermore, users operating dedicated servers, *i.e.,* seed boxes, to download and seed 24 hours a day. Those seed boxes can be rented from providers specialized in this area. Seed boxes are a gray area of concern since torrents can be added automatically, but

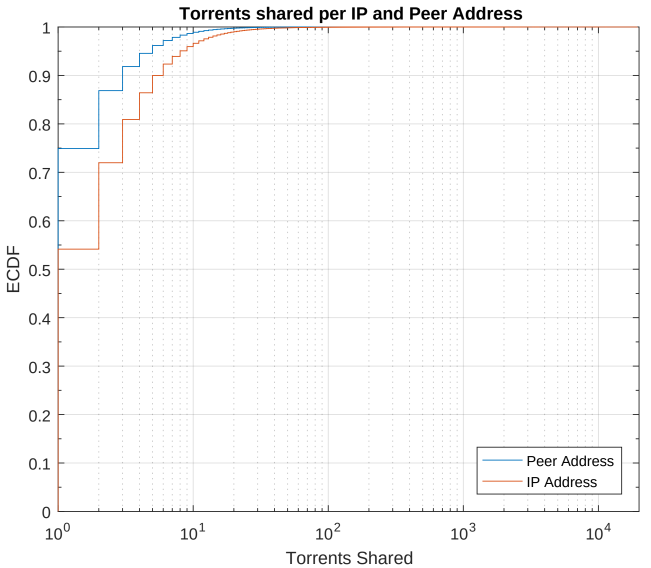
Table 5.1: Statistics describing the torrents shared per IP and Peer Address from the VIOLA 2016 data.

Percentile	5th	25th	Median	75th	95th	Mean	Max
Peer Address	1	1	1	2	5	1.75	19,528
IP Address	1	1	1	3	8	2.93	19,528

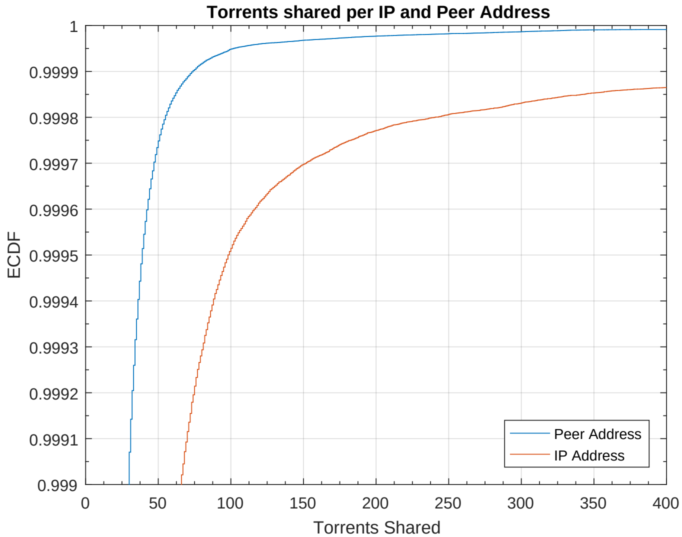
the system is operated by humans. However, in public BT trackers, it does not make sense to download and share content in which a user is not interested since there is no share ratio enforcement in place. However, seed boxes can be used to hide the location and identity of the owner, who can circumvent local anti-piracy measures. Furthermore, there is the possibility of measurement systems being at work which have to be filtered. Since those are potentially connected to all torrents measured. Measurement systems are not only operated for research purposes, but also for copy right law enforcement. To filter those, a reasonable threshold of torrents per peer or IP address needs to be defined.

Figure 5.3a presents the ECDF of torrents per peer address and torrents per IP based on data from a single day. Table 5.1 provides the summary statistics of the same data. The most interesting observation is that 95% of the peer addresses share only five torrents or less and almost 75% of them share only one. As expected, there are more torrents shared per IP address than per peer address. This is an effect of Network Address Translation (NAT) and Virtual Private Network (VPN) services, consolidating multiple users in one public IP address. However, Table 5.1 proves that at least 50% of IP addresses accommodate only a single user as the number of torrents shared is equal to the corresponding peer addresses up to the median. The mean shows a significant difference between the two peer identification schemes. Those differences lead to the conclusion that it is more accurate to identify peers by the peer address, *i.e.,* IP address and port number, than only by the IP address.

Figure 5.3b presents a detailed view of the ECDF of torrents per IP and peer address. The ECDF curve of the torrents per peer address changes direction rapidly between 50 and 100 peers. Considering that only 0.026% of peer addresses fall in the category of 50 or more torrents shared, applying 50 or even 100 as a filtering threshold will not affect the number of peers noticeably. However, depending on the network being constructed those peers filtered can have a large effect on the resulting network, since they introduce an unrealistic amount of connections. Thus, for the further analysis, 50 and 100 are used as filtering thresholds.



(a) Full Empiric Cumulative Distribution Function (ECDF).



(b) Excerpt of the ECDF.

Figure 5.3: ECDF of torrents per IP and peer address.

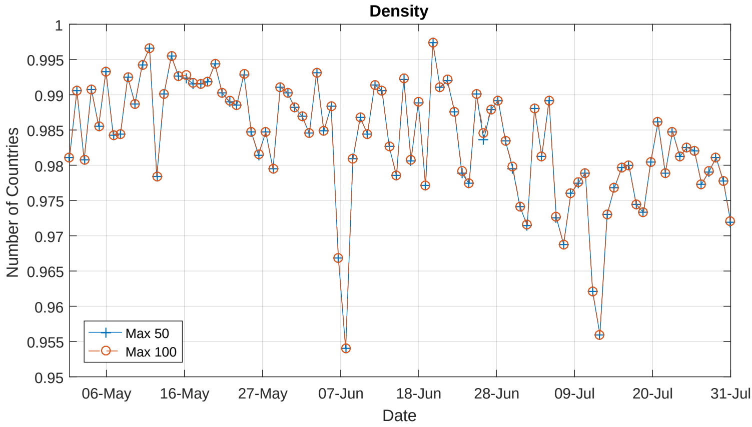


Figure 5.4: Density of the country based network.

5.2 Social Network Analysis

SNA traditionally investigates individuals and their social relations, which are modeled as graphs consisting of nodes (individuals) and edges (relations, *e.g.,* friendship) [57]. Modeling a system as a graph enables the use of several SNA measures, which indicate properties of nodes or the full network, termed node and network-centric measures respectively. With the transformation method presented herein (*cf.* Section 5.1), the VIOLA data can be abstracted into different one-mode graphs or networks, allowing standard SNA measures to be calculated on those networks. BT swarms can contain many peers and, thus, a peer-centric network can be very large, too large to compute in a reasonable time. Suitably, IP addresses can be abstracted to countries or ASes, reducing the number of nodes. However, this may result in a very dense network in which edge weights need to be taken into account.

5.2.1 The Country Network

Figure 5.4 shows the density of the country network over the three months of measurement data. Density measures the ratio of existing edges to potentially existing edges, meaning that a fully meshed graph has a density of 1. The two filter levels (50 and 100) do not affect the density of the network, meaning that no edges disappear if the filter is changed. Thus, all countries participating in BT file sharing have connections to each other that are not based on over proportional peers. The effects of measurement issues are clearly visible around June 8 and July 12. The country net density lies in the range of 0.95 to 1, showing that the network is almost fully connected. Therefore, when calculating SNA measures, the edge weight needs to be taken into account to produce meaningful results.

Figure 5.5 presents the ECDF of all edge weights in the complete dataset. The weight signifies how many torrents are commonly shared in the adjacent countries. Thus, Figure 5.5 shows that almost 50% of pairs of countries commonly share between 100 and 1,000 torrents. Furthermore, when investigating individual days, the distribution is very similar to the total distribution, rendering an in-depth analysis of daily edge weight distribution obsolete. This diversity in edge weights promises improved results when they are considered in SNA measure calculations. The edge weight is used as cost in shortest path calculations, therefore, it is necessary to alter the weights to their reciprocal value, *i.e.,* dividing one by the weight. This way, a heavy edge is considered to be shorter than a light edge.

The most interesting aspect of the country network is the importance of nodes, *i.e.,* countries, in the network. To assess this importance, several node-centric measures are available. Figure 5.6 presents the ECDF of the strength, betweenness, closeness, eigenvector, and PageRank (PR) centrality measures normalized to the interval [0,1]. The betweenness measure allows a distinct differentiation of the top nodes, however, the distinction of the nodes outside the top 5 is marginal. In the case of the country network, which is almost fully connected, paths with low cost will be on many shortest paths. Thus, countries sharing many torrents with one or more neighbor will have a high centrality. Closeness shows an adverse behavior, producing a clear distinction among the least central nodes, but very close value for the most central nodes. The strength, eigenvector, and PR measures produce similarly shaped curves, although, with different values. These measures produce the best overall distinction between nodes.

Table 5.2 shows the top 10 countries with their normalized average measures. The observation that strength, eigenvector, and PR centrality produce the best distinction, are confirmed by the raw numbers. The table allows comparing individual countries and their ranking produced by the different measures against each other. The United States of America (US) is the most central country according to all measures. This result is surprising since claims have been made that file sharing has been reduced greatly in the US and Canada (CA) [52], ranked 2nd by the betweenness

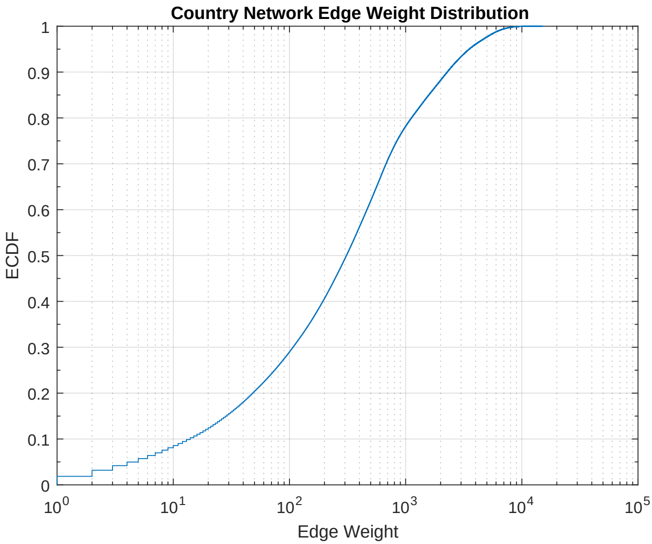


Figure 5.5: Distribution of the edge weights for all days in the measurement period.

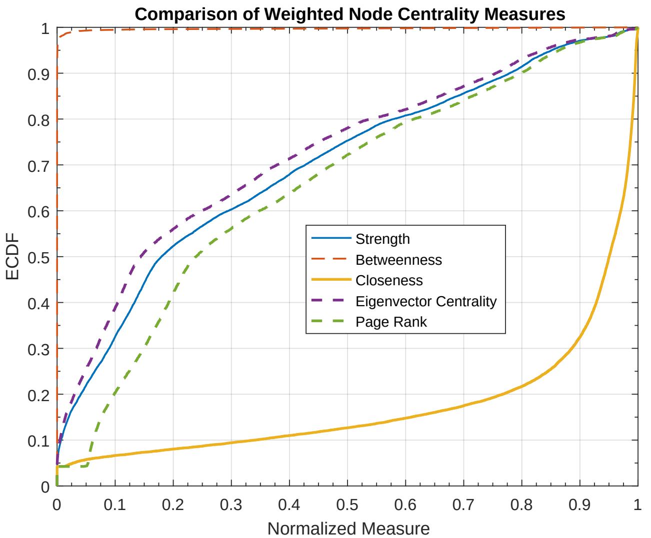


Figure 5.6: Comparison of four node centrality measures.

Table 5.2: Top 10 countries according to different node centrality measures.

Strength		Betweenness		Closeness		Eigenv.		PageRank	
US	1.000	US	1.000	US	1.000	US	1.000	US	1.000
GB	0.981	CA	0.073	GB	0.999	GB	0.979	GB	0.983
CA	0.968	GB	0.032	CA	0.999	CA	0.965	CA	0.972
AU	0.960	FR	0.011	NL	0.999	AU	0.958	NL	0.964
NL	0.960	IT	0.006	AU	0.998	NL	0.956	AU	0.963
FR	0.915	AU	0.005	FR	0.998	FR	0.906	FR	0.925
BR	0.904	NL	0.004	BR	0.997	BR	0.896	BR	0.914
IN	0.886	BR	0.004	IN	0.997	IN	0.874	IN	0.897
ES	0.870	ZA	0.002	ES	0.996	ES	0.856	ES	0.884
PH	0.867	SE	0.002	CN	0.996	PH	0.853	PH	0.877

measure and 3rd by all others. One explanation for the ranking of the US is the high diversity of its population that downloads content in many different languages, strengthening the ties to other countries. Apparently, it can be difficult to get content with a domestic audience through legal channels abroad. Thus, file sharing is the only or one of a few alternatives. All measures rank the US, Great Britain (GB), and CA among the top three. The following ranks are not as clearly defined; the Netherlands (NL), Australia (AU), France (FR), and Brazil (BR) are consistently ranked among the top 10. India (IN), Spain (ES), and Philippines (PH) are consistently ranked among the top 10 with strength, eigenvector, and PR, while closeness replaces PH with China (CN). The betweenness measure includes Italy (IT), South Africa (ZA), and Sweden (SE) instead of IN, PH, and ES.

Apparently, English speaking countries play an important role in the sharing of video content; it is also interesting to see smaller European countries, such as NL, high in the rankings of all measures; even ranked before larger Asian countries, such as CN and PH. According to the ranking, French seems to be the second most important language, explaining the high ranking of CA as it acts as a bridge between English and French speaking countries such as FR. Table 5.3 supports this theory by showing how many torrents are in the dataset of a certain language. Although there are more torrents in Hindi than in French, FR is consistently higher ranked, which can only be explained by FR having more or stronger relations with other countries.

The presented average node centrality measures give a good overview over the full dataset. However, the changes of those measures over time, *i.e.,* their evolution,

Table 5.3: Torrents shared in languages. Torrents that did not have a language available are marked N/A.

Language	Torrents	Language	Torrents
English	54,545	Italian	741
N/A	36,406	Tamil	612
Hindi	2,372	Korean	362
French	1,498	Japanese	358
Spanish	889	Portuguese	284

are an important aspect that needs to be considered. Figures 5.7-5.9 present the daily measures of the respective top 10 countries from Table 5.2.

Strength centrality (*cf.* Figure 5.7) shows a consistent ranking with only few changes, *i.e.,* lines crossing. Also, the partially missing data around June 8 and July 25 are reflected by sharp decreases in the strength of all nodes during those times. Since the strength measure mainly depends on the number of torrents shared in a country, an increase means that more torrents were shared with other countries. Interestingly, on May 16 a gap between the top 5 and the rest emerges, leading to the conclusion that torrents were added that were mainly shared among those top 5 countries. The gap starts to decrease on June 17 when FR is closing it.

Betweenness centrality (*cf.* Figure 5.8) makes a very clear distinction of the top country (US) compared to the others, even on a logarithmic scale. The US consistently show a betweenness that is an order of magnitude higher than the runner-up (GB). For the rest of the nodes, a high fluctuation of betweenness can be observed. Thus, depending on the day, the ranking of the most central nodes will look different. Since betweenness is the number of the shortest paths a node lies on, a higher betweenness for one node means a lower one for another. This split can be observed before May 15, when GB, NL, and BR were rising while AU and PH were decreasing. This could be due to a very popular content being released and first shared in the rising group, the similar time zones of those two groups support this explanation.

PR centrality (*cf.* Figure 5.9) shows a distinction very similar to the strength centrality. However, PR is less dependent on the number of torrents than strength as it shows less general fluctuation. Furthermore, also the outages are visible, but they manifest as peaks instead of troughs. Again the gap between the top 5 and the rest of the countries is clearly visible. Opposed to betweenness, PR is increasing for all

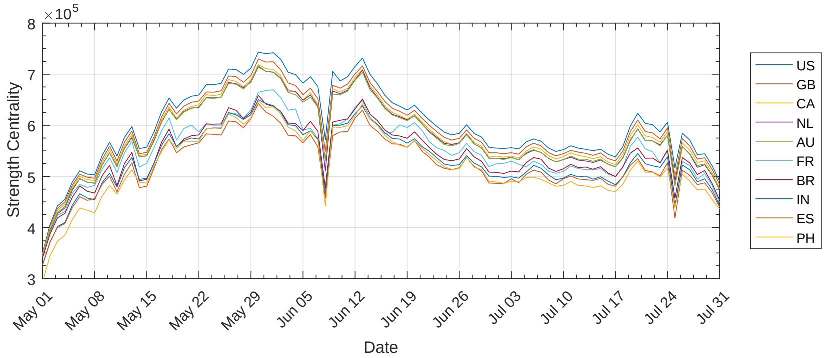


Figure 5.7: Strength centrality evolution of the country network.

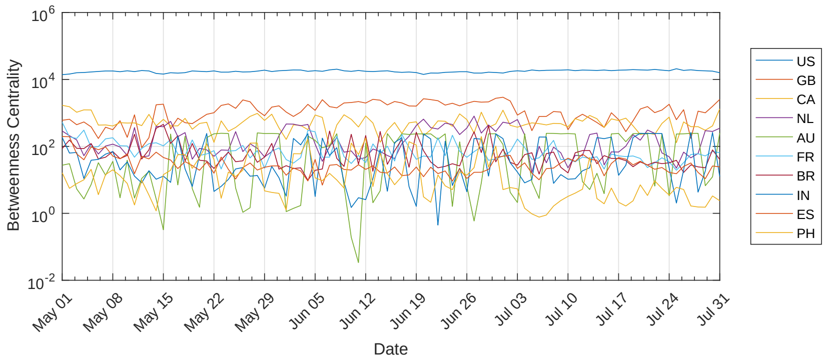


Figure 5.8: Betweenness centrality evolution of the country network.

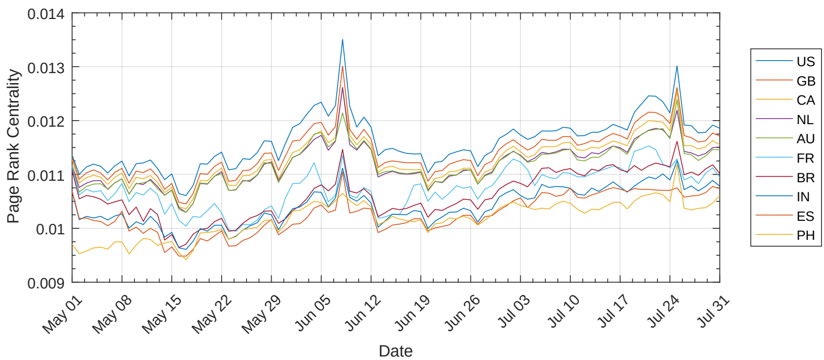


Figure 5.9: PR centrality evolution of the country network.

nodes, although, some increase more than others. All nodes displayed show small decreases after every major tick, corresponding to Sundays. This effect weakens after June 26 which coincides with the air date of the last episode of Game of Thrones (GoT) season 6.

The analysis of the country network provided an overview OF the dataset and led to the surprising conclusion that the US and CA are still crucial for the BT network. Those results directly contradict the popular opinion that piracy is reduced in Northern America [55, 2] and indicates that relative traffic numbers are not a good indicator for piracy levels. The analysis showed that traffic is highly international and, thus, it is worth to look more closely at the AS and the content shared among them.

5.2.2 The Autonomous System Network

To provide a deeper insight into the locality of BitTorrent swarms, peers were mapped to their ASes. This mapping is exact, since ASes have IP prefixes assigned. Compared to the country network, the AS network is more accurate, but it lacks the geographic aspect because a single AS can be used all over the world. Additionally, the edges, representing torrents, can be weighted according to the size of the files and the number of peers downloading it. Thus, the edge weight signifies the amount of traffic, which two ASes potentially exchange.

The weight of an edge is best calculated in stage 2 when all the ASes sharing a torrent are combined. According to the BT specification [13], peers are selected randomly, meaning any peer has equal chances of downloading from another peer. Thus, the potential amount of traffic flowing to ASj can be calculated by multiplying the number of peers with the size of the respective torrent. The amount of traffic coming from a specific AS is proportional to the portion of peers from the swarm located in said AS. For illustration, the formula to calculate traffic from ASi to ASj, $T_{ASi-ASj}$, for a torrent is given in Equation 5.1.

$$T_{ASi-ASj} = Peers_{ASj} \cdot FileSize \cdot \frac{Peers_{ASi}}{Peers_{total}} \tag{5.1}$$

Equation 5.1 shows that the traffic between ASes is symmetric, since switching ASi and ASj will not change the result. This property is important for the performance of edge weight calculation; it means that the weight has to be calculated only

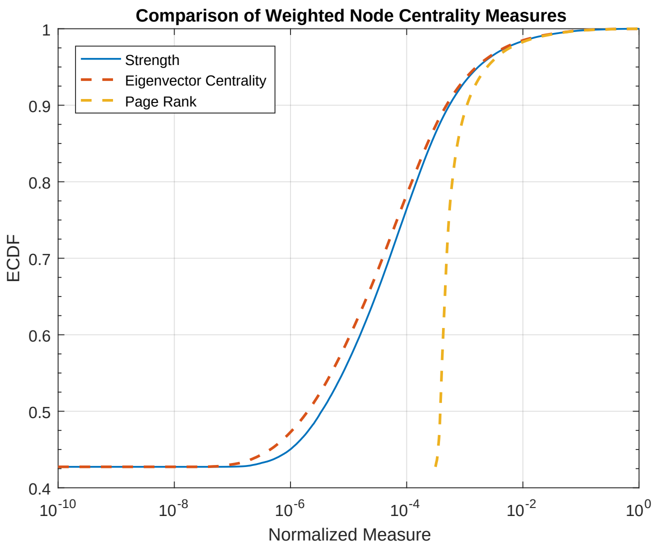


Figure 5.10: Comparison of four node centrality measures.

once per pair of ASes. Semantically, this weight reflects the traffic uploaded from one AS to the other AS, which is the same amount traffic uploaded by the second AS to the first.

The resulting network represents ASes connected by their potential data flow in one direction. The weight does not reflect the actual traffic exchanged, as this would require more information on peers' bandwidth and also the completion of downloads. However, the goal of this analysis is to compare the relations of ASes for which this weight provides a suitable indicator.

Figure 5.10 presents strength, eigenvector, and PR centrality as their ECDF. For this network calculating shortest path based metrics, *e.g.,* betweenness and closeness, is not feasible, since the network averages at 17,000 nodes and 61 million edges a day. Thus, the complexity to calculate all the shortest paths is too high. Similar to the country network, the three remaining measures produce similarly shaped ECDF curves. The eigenvector is very close to the strength while PR shows a steeper curve and a clear cut-off at about 42% of the values, for which no distinction can be made. Thus, for this network the three measures exhibit similar behavior and are, thus, interchangeable.

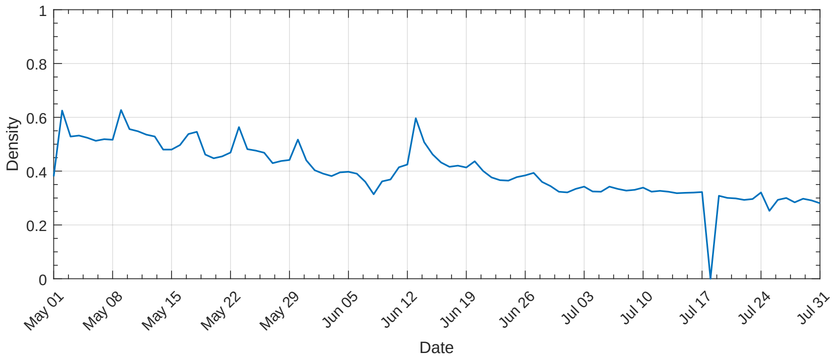


Figure 5.11: AS network file size weighted density evolution.

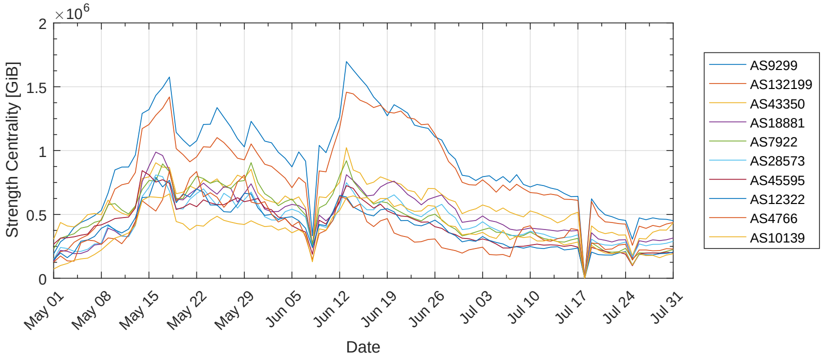


Figure 5.12: AS network file size weighted strength centrality evolution.

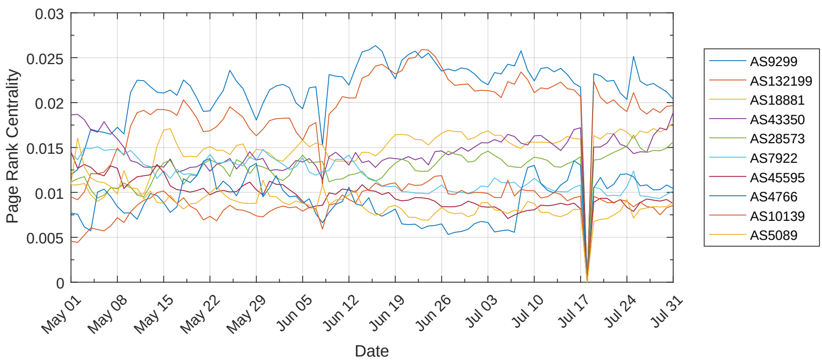


Figure 5.13: AS network file size weighted PR centrality evolution.

Table 5.4: Top 10 AS ranked according to strength and PR centrality.

Rank	Number	Organization Name
1	9299	Philippine Long Distance Telephone Company
2	132199	Globe Telecom Inc.
3	43350	NForce Entertainment B.V.
4	18881	TELEFÔNICA BRASIL S.A
5	7922	Comcast Cable Communications, LLC
6	28573	CLARO S.A.
7	45595	Pakistan Telecom Company Limited
8	12322	Free SAS
9	4766	Korea Telecom
10	10139	Smart Broadband, Inc.

Figure 5.11 depicts the density of the network. The density is the percentage of possible edges that actually exist. This figure shows distinct peaks at the beginning of every week, which have been observed before, meaning that certain content can connect ASes, which would otherwise not share any common content. It also shows that the release date of that content is more important than the time of the week. Apparently, BT users want the content as soon as it is available, indicating that the model of releasing content graduated over the different regions increases piracy. The density is decreasing over time which is explained by the overall reduction in torrents over the measurement period.

Figure 5.12 presents the strength evolution of the top 10 ASes. The strength is the sum of all edge weights of an AS, meaning it is the aggregate of the potential traffic coming in or out of the AS. Table 5.4 provides the names to the AS numbers in the legend. Surprisingly, the two top ASes are located in the Philippines, which were only ranked 10th or higher in the country network. Thus, the Philippines share fewer torrents but create more traffic with those torrents due to having more peers sharing them. Furthermore, there is a large gap between the two leading ASes and the rest between June 12 and June 26., exactly until the last episode of GoT Season 6 aired.

Figure 5.13 shows the PR centrality for the top 10 ASes. The ranking is slightly different; however, the two top ASes from the Philippines are still in the top two ranks. Furthermore, the gap in strength opening just before June 12 is also observable in the PR evolution. PR also shows smoother peaks than strength. Furthermore, PR shows some distinct peaks after June 16 which are not as well visible in the strength figure. Those peaks are not as regular since they represent individual movie torrent releases, which typically spawn a flash crowd and then start to lose popularity slowly.

The AS network allows to compare ASes according to the traffic they are producing. Although the method does not produce accurate traffic numbers, it shows which ASes share more data. It is surprising to see 2 Philippine Internet Service Providers (ISPs) on the top of this ranking. Explanations for those observations are the liberal legislature regarding online piracy and the presence of VPN gateways. Another unexpected appearance in the top 10 is the Dutch company NForce Entertainment B.V., specializing in hosting of dedicated servers with fast Internet connection. These are the requirements of seed boxes that are used solely to download torrents. For public BT trackers, users typically select this option if the legislature in their country does not allow file sharing, effectively avoiding prosecution. For the analysis of content piracy this means that a significant part of downloads can not be localized as the true identities of those file sharers are hidden. However, it shows that file sharers are resourceful and are even willing to pay for solutions. If users are willing to pay to illegally share content, this must be the only option to access content or they would just buy it. Another motivation to rent seed boxes is the enforcement of share ratios in private BT communities. However, this is not relevant for the results presented herein, since no private communities were included in the measurements.

5.2.3 The Torrent Network

Providing another perspective into the BT data, a network of torrents is created, representing the right-hand part of Figure 5.1. Thus, this network will connect torrents, *i.e.,* videos, by the number of peers that share them commonly. A network like this is the basis for content recommendation as it can be used to discover content starting from one content a user has consumed. The number of common peers among two torrents are used as the edge weight, signifying the strength of the connection.

Figure 5.14 provides the ECDF of node centrality metrics over the full measurement data, giving an overview of the different measures. There is no noticeable difference when the filter threshold is changed from 100 to 50. Thus, only a threshold

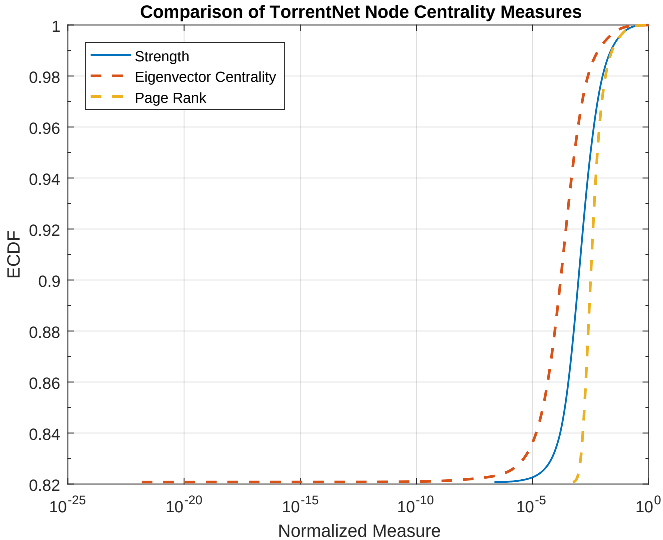


Figure 5.14: Comparison of three node centrality measures of the torrents' network with peers sharing 100 or less torrents.

of 100 torrents per peer is used. The ECDF shows that 82% percent of the torrents have no centrality at all with all metrics used. Thus, only 18% of the measured torrents are relevant for many users. This confirms the observations made regarding popularity (*cf.* Section 4.2.2), showing a near exponential popularity distribution. Furthermore, the strength, eigenvector, and PR measures show a very similar distribution. Only the eigenvector is exhibiting a long tail covering much lower values than the other two.

The density of the networks with a filter threshold of 50 and 100 are presented in Figure 5.15. The difference between the two filter levels manifests in a slightly lower density for the stricter filter of 50 torrents per peer. However, the shape of the curves is very similar and also the technical difficulties become visible around June 8 and July 19. At the beginning of the measurement, density is above 20%, but it decreases quickly and then remains below that threshold. The decline in density is explained by the lower number of torrents at the start of the measurement, but due to the near exponential popularity and the quadratic growth of potential edges, new torrents bring fewer connections on average than the existing part. Thus, the density is dependent on the number of torrents in the network.

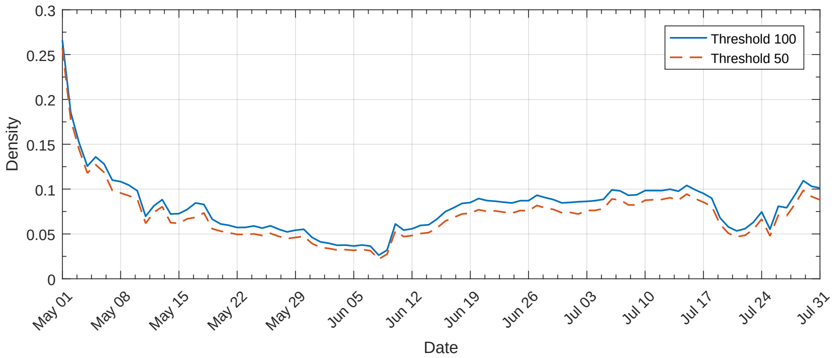


Figure 5.15: Torrent network density evolution.

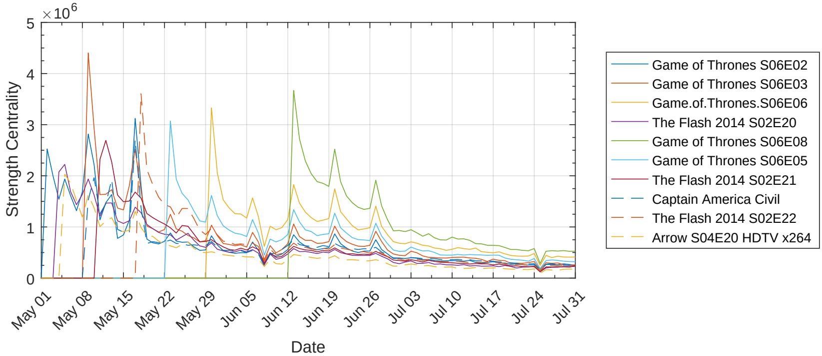


Figure 5.16: Torrent network strength centrality evolution with filter threshold of 100.

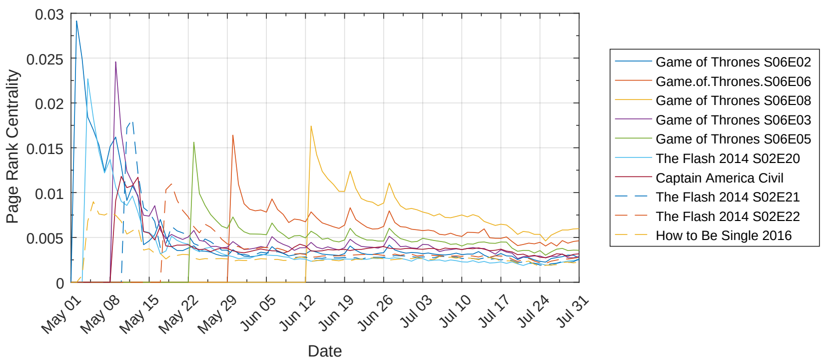


Figure 5.17: Torrent network PR centrality evolution.

Figure 5.16 depicts the top 10 torrents' strength evolution for a filter threshold of 100 torrents per peer. The top 10 ranking is done on the average strength over the full dataset. Apparently, the most relevant content measured were the Television (TV) shows GoT and "The Flash", but also the movie "Captain America" is among the top 10. The GoT episodes exhibit a very distinct weekly pattern; whenever a new episode is released, a peak in strength appears the day after (due to the UTC used in the measurement). Furthermore, each of those peaks is accompanied by smaller peaks of previous episodes. Those smaller peaks have two reasons: First, users shut down their BT client after an episode is downloaded, and restart it when a new one comes out and continue seeding the old episodes. Second, users are reminded to download the old episode by the new episode coming out soon. The second reason also explains the slight increase in old episodes before the day of the release of a new episode.

The top 10 torrents are presented in Figure 5.17 according to PR centrality. The most notable change to the strength ranking is the distinct group of GoT episodes on top of the ranking. Episodes of "The Flash" are consistently ranked lower, and "Arrow" is exchanged with the movie "How to Be Single". Another difference can be observed after March 15 where there is no high peak compared to the other weeks, although, the strength centrality shows at least peaks for "The Flash" and "Captain America". GoT episode 4, which was released on March 15, is missing from the list, which is surprising as episodes 3 and 5 both are in the top 10. This must have to do with "Captain America" becoming available before and, thus, interrupting the network. Therefore, it can be concluded that both network measures, strength and PR, are a relative measure of popularity, contrasting the standard measure of total downloads or views. Appendix A presents the seeder and leecher numbers for the top movies and TV shows, which confirm the observations of the relative nature of SNA measures. Furthermore, the comparison between movies and TV shows shows that TV shows gain and loose popularity more quickly than movies. Under this aspect, the interruption of GoT episode 4 is even more noteworthy.

The torrent based network abstraction gives a different angle on the BT data compared to a peer-centric approach. The analysis and comparison of SNA measures on the torrent network have shown their usefulness when comparing content against each other. Applied on the VIOLA data, the strength and PR centrality give an indication which content is most important on a given day. Observing torrents' centrality, distinct weekly patterns were discovered. Surprisingly, those weekly patterns also appear in older episodes of TV shows and even movies are affected.

5.3 Chapter Summary

The abstraction of BT data into user and content networks proved to be a feasible method for the analysis of country influence, relative traffic shares of AS, and relative content popularity. The method presented in 5.1 proved to apply to a dataset of 4.7 TB.

The country network analysis leads to the conclusion that English-speaking countries (US, GB, CA, AU) are the most influential as they share the largest number of torrents across borders. Finding the US clearly on top of this list is surprising since the laws are strict and VOD services offer the largest catalogs. The number and diversity of the US population can, at least partially, explain this effect, since this means that much content in foreign languages is shared in the US, especially if that foreign content is not available from any legal service.

Furthermore, the country network has shown that English-speaking countries (US, GB, CA, AU) are the most influential as they share the largest number of torrents across borders. However, diving deeper into the traffic weighted AS network showed that two Philippine ASes consistently shared the largest amount of data over the full data analyzed. Finally, the torrent network gave insight into the type of content that is most popular and revealed patterns in the way BT is used.

Diving into the file-size-weighted AS network showed that two Philippine AS consistently shared the largest amount of data over the full data analyzed. The third position, “NForce Entertainment B.V.”, was also surprising as the AS belongs to a dedicated server hosting company rather than an ISP, questioning the claims of reduced piracy in Northern America and documenting the importance of BT measurements to assess anti-piracy measures.

The torrent network showed that the most popular content during the measurements were the “Game of Thrones” episodes, but also other TV shows and some movies were among the most influential. This popularity of TV shows leads to distinct weekly patterns due to users downloading new episodes immediately after their release. Movies behave differently and do not seem to be influenced by the TV shows.

Those results show a level of detail over an extended period, which has not been seen before due to the lack of detailed datasets (*cf.* Section 2.5). This level of detail is needed, for instance, to compare content released independently from each other, *e.g.,* GoT and "Captain America", and the effect those releases have on the attention of users. The shared-together property of the torrent network provides the basis for producing recommendations for users, even outside the BitTorrent system.

The findings presented herein confirm the relevance of detailed BT data that VIOLA can deliver. VIOLA provided a dataset of unprecedented detail and completeness, allowing the investigation of the relation of content and users over time. With previous datasets, this was not possible since at least one of those three dimensions lacked detail. Therefore, VIOLA provides data of unprecedented quality and completeness.

To be suspicious is not a fault. To be suspicious all the time without coming to a conclusion is the defect.

Lu Xun

6

Summary and Conclusion

This Thesis has shown that BitTorrent (BT) is still widely used and that the most popular content shared is copyrighted and, thus, shared illegally. Therefore, detailed and complete BT measurements are required to assess measures and trends, such as the use of rented servers. Furthermore, this data collected is beneficial for a multitude of purposes, *e.g.,* content recommendations. The comparison of prior BT measurement studies and their resulting data sets showed that a gap in previous measurements was identified. Specifically, measurements, or data sets, providing multiple samples per hour, including tens of thousands of swarms, and covering months were missing. Even if the sampling rate is ignored, the studies including more than 10,000 swarms typically provided only a single sample not reflecting the time dimension well. However, it is this time dimension that is key in building the factual basis to assess changes in the file sharing behavior, *e.g.,* due to new laws or the confiscation of BT indexes. For those purposes, data of high resolution is required, but also over an extended period. Therefore, BT is still very relevant and prior research did not cover it in sufficient detail.

6.1 Summary of Contributions

This thesis made the following contributions: (a) the design and implementation of a distributed scalable BT measurement system, termed VIOLA, which can collect data sets that fill the identified gap; (b) a dataset covering 3 months, 70,000 swarms, and 3 samples per hour, proving that the VIOLA system works as designed and providing the basis for novel insights in BT behavior; (c) a method to transform the measurement data into graphs with different semantics building the basis for quantifying user and content-centric properties of the BT system.

6.1.1 BTPC and VIOLA

To accurately measure BT swarms, the BitTorrent Peer Collector (BTPC) problem, *i.e.,* collecting all peers of a swarm, was solved. First, swarm size estimators, *i.e.,* simple and maximum likely hood estimators, were investigated through simulations. Second, the collection of a swarm through random draws was modeled accurately to orchestrate the collection across multiple collectors. Analysis of real world tracker responses confirmed that both estimators provide accurate swarm size estimates and that trackers report accurate swarm sizes. By applying a sliding window to the estimation, accurate swarm sizes can be continuously estimated. The Mainline Distributed Hash Table (MDHT) provides more unique peers per response than any of the regular trackers, but due to timing parameters, chosen by its developers, the estimations fall short. Furthermore, more peers were found to be in the MDHT than in any of the investigated trackers. This analytical model, derived to predict the number of tracker requests and needed to collect a swarm, allows orchestrating the collection of a swarm among multiple collectors.

Based on those insights, the decision was taken to follow a distributed approach for the design of Video Consumption in Overlay Networks (VIOLA), employing a master-slave architecture. While the slaves are responsible for querying trackers and the MDHT, the master discovers new torrents and persists the results received from the slaves, allowing the system to scale horizontally should slaves become overloaded. By dividing the collection into different categories, also the master can be scaled horizontally. A relational data model for the data that can be collected from trackers and the Distributed Hash Table (DHT) was developed and transformed into a schema for MySQL [50] and Avro [61]. Measurement runs have shown that the MySQL based storage reaches its limit rather quickly and, thus, the Avro approach was chosen for large-scale measurements. The analytical collector model was implemented in the adaptive scheduler, leading to an efficient collection of large and small swarms with negligible communication overhead.

The investigation of the BTPC has led to a scalable and autonomously orchestrated systems design. The data is written to a flexible and standardized output format, which can be easily transferred to data storage or processing systems such as Hadoop. The resulting implementation fulfills the requirements to close the gap in BT measurements identified in Chapter 2.

6.1.2 VIOLA Measurements

To assess the capabilities of the VIOLA implementation and to collect a dataset that goes beyond the state of the art, two measurement runs were executed: in April 2015 and from May to July 2016. The 2015 measurement was executed with a simple scheduler not considering swarm size for collection, resulting in a dataset covering 14 days and 5,000 swarms. The maximum swarm size that could be fully collected with the simple scheduler and those parameters chosen was identified to be at approximately 10,000. Furthermore, the limited usability of traditional SQL databases for this specific use case was discovered. Both of those lessons learned were addressed by the implementation of the Avro-based storage component and the adaptive scheduler. With the improved VIOLA system, three months of measurements were collected. The result was that 98% of the swarm collections (3 times an hour), collected more than 95% of the swarm size reported by trackers. Considering that some swarms contained well over 100,000 peers and that the infrastructure used was comparable, this is an excellent result and delivers enough accuracy to analyze the data in detail.

Compared to previous BT measurements, the VIOLA dataset provides a new quality by combining the time, content, and user dimensions in an unprecedented way. Chapter 2 categorized the VIOLA dataset along the same criteria as the prior measurements (*cf.* Figure 4.15) and showed that VIOLA fills the gap in BT measurements. Also in terms of time frame, VIOLA provides new results, covering months. Thus, VIOLA enabled analytical methods as presented in Chapter 5.

A first analysis of the collected data showed that North America has fallen behind Asia and Europe in the total number of active peers compared to the Ono study [51]. The popularity of content, measured by maximum swarm size, shows an almost exponential distribution. Tracker reported swarm sizes are lower than those collected from the MDHT, confirming the initial results from Chapter 3 on a large scale. An anomaly was observed in the rank distribution of tracker reported swarm sizes, visible over the full data and specifically when looking at the July data. More detailed investigations of tracker reports show a new tracker emerging and seemingly reporting unrealistic numbers. This event coincided with the seizing of the "Kickass Torrents" portal's servers by the FBI, showing that a single measurement snapshot can be very biased and continuous measurements of BT swarms and sufficient level of detail, as provided by VIOLA, are necessary to draw valid conclusions.

6.1.3 Analysis

To analyze the collected data and quantify the importance of individual countries and Autonomous Systems (ASes), a method to transform the tracker responses into a one-mode graph, allowing quantification of node centrality was introduced. The presented method applies to any trace that contains a user Identifier (ID), content ID, and time. If the users can be mapped to a location or Internet Service Provider (ISP), the same networks and metrics as presented in this thesis can be produced, making different data sets comparable. Two options for projecting the BT network, consisting of torrents and peers, exist.

The first option consists of replacing the torrent nodes with edges between the peers, which were aggregated to countries and ASes with edges weighted according to the potential traffic exchanged with other ASes. The country network showed a very high density, being consistently above 95%. Thus, almost every country is directly connected to almost any other country. As a consequence, edge weights (number of commonly shared torrents) were important when calculating node centrality measures. The different Social Network Analysis (SNA) measures applied, led to different results. However, the results were similar having 9 (except betweenness having 7) out of the top 10 in common with the others. The US was the most central country in all metrics followed by GB and CA. Despite efforts to eradicate content piracy, the US and CA are important in the BT ecosystem. However, the AS network weighted by the potential traffic shows that the Philippines are responsible for most traffic during the whole measurement period. Furthermore, a Dutch provider of servers is among the top ranked AS. The only explanation is the use of this provider's servers as seed boxes, acting as a proxy for users that cannot run a system

24 hours a day or that are afraid to use their Internet connection to share pirated content. Thus, it can be concluded that raw traffic numbers are not sufficient to assess the degree of piracy in a region. Furthermore, if piracy is reduced in one place, it might rise in another due to increased use of seed boxes or Virtual Private Network (VPN) services.

The second option is to replace peers with edges between torrents, resulting in a content centric network. This network can be used to discover or recommend new content to users by following the edges of content they already consumed. Those recommendations are not restricted to BT, but can be used for any Video on Demand (VOD) service. The content network also allows to rank content according to its centrality in the BT system through node centrality measures, providing a new angle on the popularity of content. The application of those measures to the VIOLA dataset revealed strong weekly patterns, caused by the release of new episodes of the "Game of Thrones" Television (TV) show. Additionally, the release of a new episode caused a peak in the centrality of older episodes and also popular movies such as "Captain America Civil War". Thus, weekly patterns are a result of releases rather than a consequence of the day of the week.

The network analysis presented showed how the noisy and redundant data from the VIOLA measurement system was transformed into a graph structure which can then be used to calculate metrics. Besides the benefit of providing information on the relations between countries or ASes and between content, SNA measures are a way of monitoring a system over an extended period. Providing the basis to evaluate changes in the BT system, *e.g.,* due to anti-piracy measures.

6.2 Review of Research Questions

Based on those contributions made in this thesis, the four main research questions and their sub-questions (*cf.* Section 1.4) can be answered. Those questions are reviewed, and the answers to them are provided.

1. What is state of the art in BT measurement and monitoring?

    (a) What are the most critical measurements and what are their conclusions?

    (b) Where is the gap in existing BT measurement methodology?

Before VIOLA, BT measurements mostly provided snapshots of large numbers of swarms or time series of a few swarms. The methodology used was mainly tracker and portal scraping, but also more advanced approaches, exploiting the Peer Exchange (PEX) mechanism, were used before. The most important measurements were the Grp. [24] providing a time series of 16 swarms comprising 440 samples in 88 hours. The second notable example is [76] providing one sample of more than 1 Billion swarms in 12h. Those two measurements represent the extremes, but in the middle of those was a gap. Furthermore, those measurements covered a maximum of 4 days, which is not enough to identify weekly patterns.

2. How can this gap in BT measurements be closed?

    (a) How can all peers sharing a file be identified?

    (b) How can a BT measurement system be built?

The analysis of the BTPC problem has resulted in a model which was used to orchestrate the distributed collection of BT swarms. Investigations of tracker and DHT time series have shown that the MDHT can be used to collect swarms faster than with trackers only. The most important piece of the engineering of VIOLA was the adaptive scheduler which implements the BTPC model and is capable of collecting swarms as large as 300,000 peers, the largest swarm that occurred during the measurements.

3. Can BT be measured accurately?

    (a) What accuracy can the designed system achieve?

    (b) What resources are required to achieve a certain accuracy?

    (c) Does the collected data confirm prior findings?

Yes, VIOLA has shown that it is possible to collect swarms accurately. Furthermore, most swarms are exactly as big as advertised by trackers and could be collected completely. Other swarms are dispersed among several trackers, but the MDHT provides far more peers than the busiest tracker. Even for the largest swarm, VIOLA has shown that it can collect them completely. The resources used in the VIOLA 2016 measurement were a server with 64 GB ram and ten slaves with 8 GB RAM; those were able to collect all swarms discovered. RAM has shown to be the critical resource in this process. Finally, when compared to prior analysis, similar characteristics could be identified. The popularity distribution of content has not changed and is consistent with prior findings. Furthermore, daily patterns over different continents are still similar, but the distribution of peers among continents has changed. Finally, the common assumption of selfish peers was clearly contradicted by the number of seeders in a swarm constantly growing.

4. What insights can be gained with such a dataset?

   (a) How can the raw measurement data be transformed to apply standard methods to it?

   (b) Which countries are important in BT

   (c) Which ASes are creating the most traffic?

   (d) Which content has the largest influence on the users?

The dataset can provide all the insights prior measurements provided, but due to its high resolution and extended period, it allows more investigations. However, this larger dataset required scalable methods of data processing, for which Spark [64] proved to be perfectly suitable to transform the raw data into different graphs for SNA. This analysis has shown that the US, despite its strict anti-piracy legislature, is the most important country in BT file sharing. However, considering file sizes, the analysis revealed that the Phillipine ISPs are responsible for the largest amount of traffic, but also a Dutch dedicated server provider was prominently placed in that ranking. Finally, TV shows emerged as the most influential content in the BT system, with the largest flash crowds and largest swarms in general. Movies show a less steep rise in downloads after their release, but the interest in movies is more sustainable than in TV shows.

6.3 Conclusions

BT is a dynamic system with a scale of those dynamics ranging from hours to years. Therefore, also the resulting Internet traffic follows those dynamics, which are influenced by factors such as natural human behavior (day and night patterns), release times of content, legislature, and the availability of VOD services. The ability of VIOLA, to continuously measure BT in short intervals, allows to capture those dynamics over the full scale.The BTPC analysis made this possible and can be re-used in any distributed collection with random samples, *e.g.,* , monitoring of animal populations in vast areas. The resulting detailed data allows to identify reoccurring patterns, due to TV show releases, which make Internet traffic more predictable and, thus, supports ISPs' network management efforts. Furthermore, comparing measurements with a year in between them, showed a move of peers from Europe to Asia, which is a long-term effect that can only be captured by repeated or continuous measurement over years.

The shift BT of users to Asia shows that content piracy is a global problem since pirates can easily circumvent legislature by renting a server abroad or by using a VPN service. While it is certainly true that some pirates moved to legal offerings when those became available, still a diverse crowd of file sharers remains in those regions. Thus, just relying on traffic measurements falls short as it mainly shows the growth of VOD services in some areas, but cannot capture the evading measures Peer-to-Peer (P2P) users take to avoid prosecution. The current discussion of piracy and counter-measures lacks this global and technical perspective. However, the results presented in this thesis can provide such a perspective, which has shown that file sharers make use of seed-boxes and VPN services.

Covering a wide selection of content allows for distinct analysis of content popularity in different regions. Thus, the VIOLA 2016 is a valuable resource for content or VOD providers to optimize their services. Better VOD services are a way of reducing piracy and improving the revenue of providers at the same time.

Finally, this thesis concludes that file sharing remains popular and a problem for copyright holders and associated industries. The new measurement and analytic approaches contributed by this thesis, constitute the tools to monitor BT, the largest P2P file sharing system, and assess measures taken to prevent piracy. Furthermore, the traces collected constitute a unique dataset for content recommendations, since the diversity of content available in BT surpasses those of VOD services, which are constrained by licensing. Thus, this thesis provided a scalable BT measurement methodology, a new type of dataset, and analyses, giving novel insight in the BT system and its users' behavior.

The investigation of the BTPC problem, the design and implementation of the VIOLA measurement system, the collection of the VIOLA dataset, and its transformation and analysis were important steps for BT measurement methodology and analysis, expanding the state of the art. However, open research and design questions remain.

Currently, SNA libraries are limited to run on single machines. Distributed frameworks, such as Spark GraphX, allow only a very limited set of methods to calculate SNA measures in a cluster. Specifically, distributed shortest path calculations are missing. Researching distributed algorithms for those measures potentially increases the scalability of SNA in general and would significantly improve the level of detail of peer-centric networks analysis.

The new characteristic of the VIOLA 2016 dataset enables a vast amount of options for analysis, going beyond the ones presented in this thesis. One option is the creation and evaluation of recommendation systems, which can then be used in VOD services or for prefetching of BT content to reduce peak internet traffic. Another option is the analysis of patterns in the data using machine learning or pattern recognition techniques, which can analyze the full dataset in much more detail than the analysis presented in this thesis.

This thesis investigated public BT usage. However, due to the risk of prosecution, Internet pirates have formed private communities, using trackers that are not publicly available. To extend the scope of BT research even further, those communities need to be considered in more detail. A first step would be to collect information on existing communities to gain an understanding of how important those are compared to the public network. In a second step, those communities need to be included in measurements to investigate the differences in content availability and user behavior.

The analysis of the VIOLA dataset has shown daily and weekly patterns. To investigate even larger time frames, longer measurements are required. Ideally, continuous measurements can be handled. While the measurement system can handle this, the question of data aggregation and management, *i.e.,* when to delete data or what aggregation level to store in the long term, needs to be answered. With a continuous measurement and analysis, long term effects, *i.e.,* changes in legislature, can be observed and anti-piracy measures can be evaluated. Additionally, the measurement itself can be extended to allow more detailed traffic estimations, requiring at least a sampling of download completion ratio per AS. Thus, the potential traffic can be substantiated to future traffic, providing a more realistic metric. Leading to the question if it is possible to create a model of Internet traffic caused by BT file sharing without the need for measuring traffic.

A

Seeders and Leechers

Trackers report the number of seeders and leechers that are in a swarm. Seeders, having completed their download, have no reason to stay in the swarm and continue to upload. Section 4.1 presented a case of the 2015 dataset which showed that with time, the number of seeders becomes larger than the number of leechers and that the number of seeders will constantly stay larger. To provide a broader view and to prove that this altruistic behavior can be observed in different swarms, more examples are given here. Thus, examples of TV Shows and Movies are presented showing the first week of the torrents' lifetimes.

A.1 TV Shows

Figures A.1 to A.3 show seeder and leecher numbers for the Game of Thrones episodes 3, 5, and 8, as those were the three biggest swarms at their peak. All three episodes exhibit a flash crowd behavior immediately after their releases. Interestingly, the tipping point in the seeder leecher ratio, *i.e.,* when seeders become more numerous than leechers, appears a few hours after the release. The length of the videos certainly has an influence on this time as an episode is typically shorter than one hour the files are smaller than movie files.

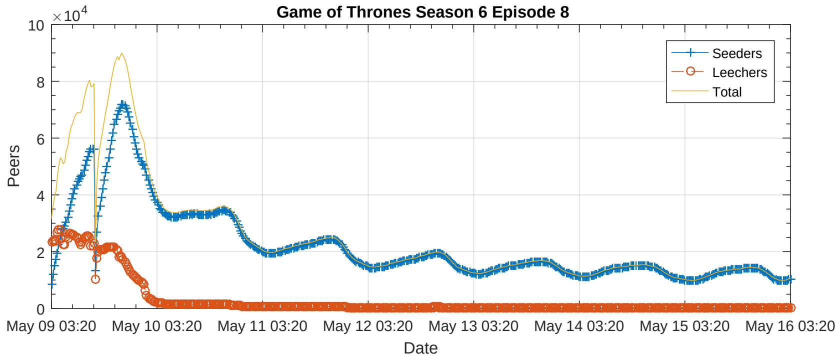


Figure A.1: Seeders and leechers in the first week of Game of Thrones Episode 3.

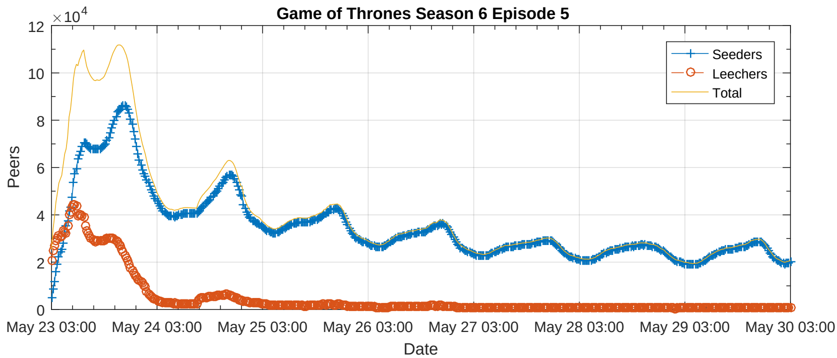


Figure A.2: Seeders and leechers in the first week of Game of Thrones Episode 5.

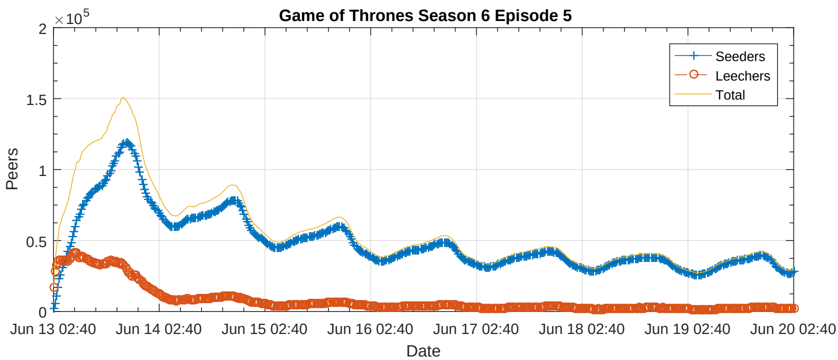


Figure A.3: Seeders and leechers in the first week of Game of Thrones Episode 8.

A.2 Movies

Figures A.4 to A.7 present the seeders and leechers for the four largest movie swarms. The flash crowd effect is still visible for Captain America and War Craft but it is not as strong as with Game of Thrones. Also, the tipping point is reached later. "Captain America 1080p" and "10 Cloverfield Lane" show less of a flash crowd behavior and the tipping point for "Captain America 1080p" is only reached after four days of its lifetime, supporting the argument with the file size which extends download times.

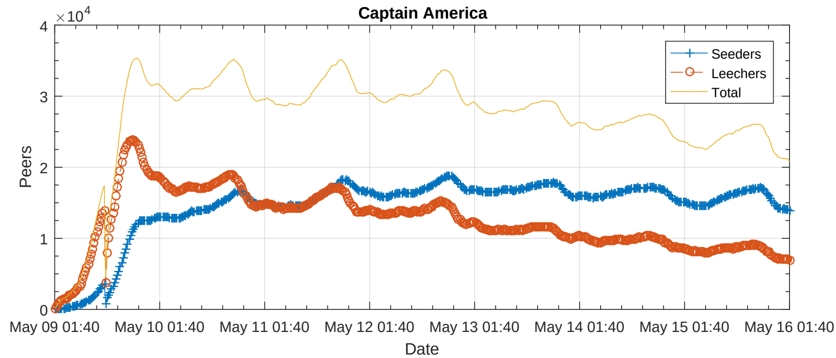


Figure A.4: Seeders and leechers in the first week of Captain America.

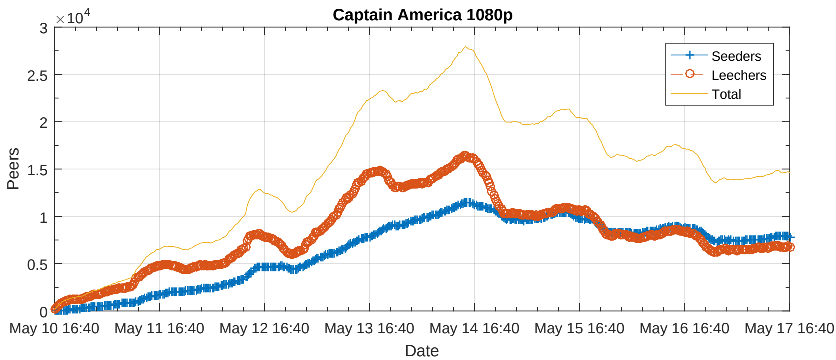


Figure A.5: Seeders and leechers in the first week of Captain America in higher, 1080p, quality.

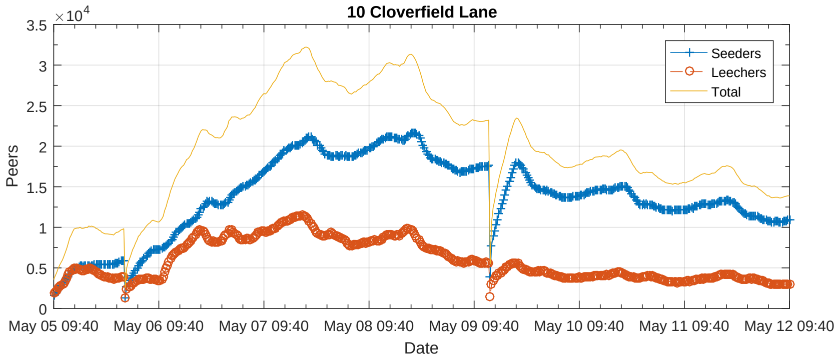


Figure A.6: Seeders and leechers in the first week of 10 Clover Field Lane.

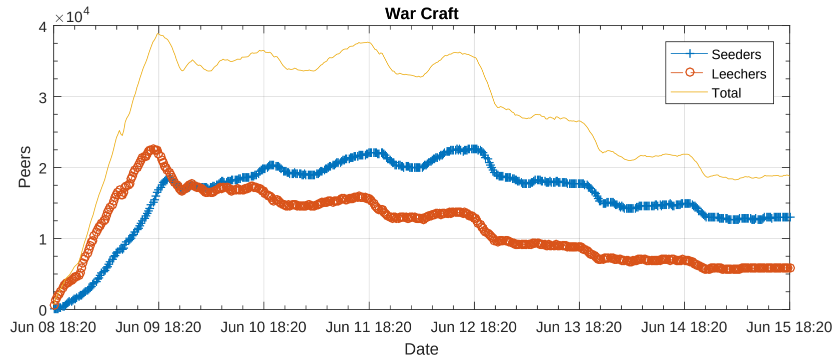


Figure A.7: Seeders and leechers in the first week of War Craft.

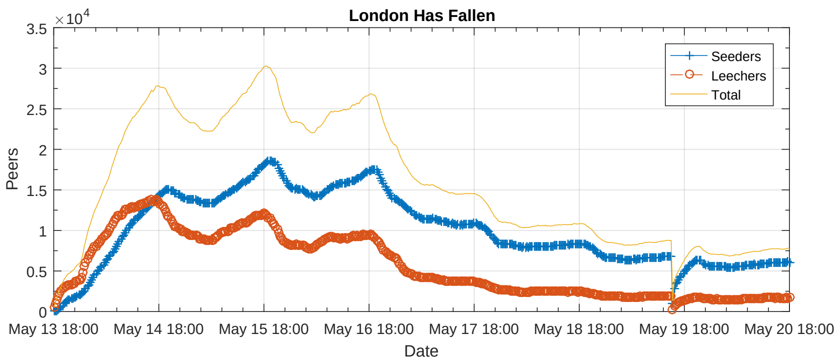


Figure A.8: Seeders and leechers in the first week of London Has Fallen.

B

Peers Measured per Hour

To give a long term view of the peers per continent, the plots of unique peers per hour are provided here for the full measurement period. Figures B.1 to B.13 show one week of measurement data. The scale of the y-axis is kept the same for all figures. Table B.1 provides the abbreviations used for the continents in the figures.

Table B.1: Continent abbreviations.

EU	Europe	NA	North America
AS	Asia	SA	South America
AF	Africa	OC	Oceania

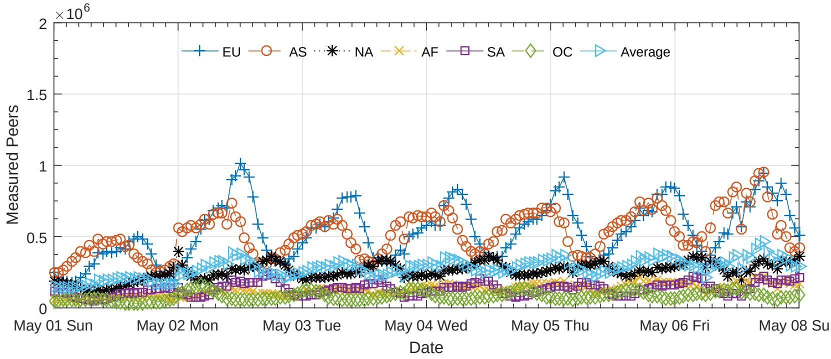


Figure B.1: Peers per continent in week 1 of the 2016 measurement.

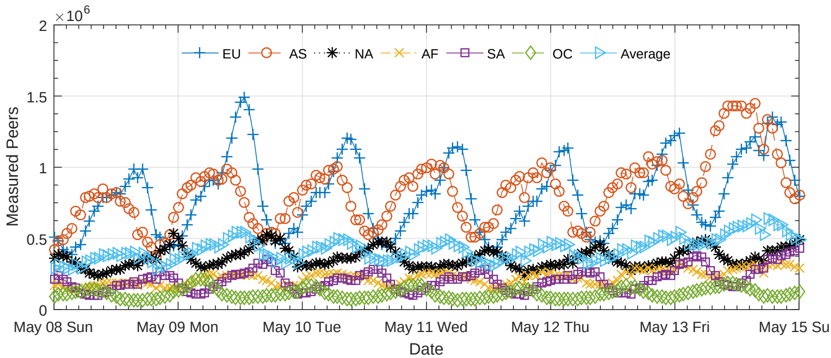


Figure B.2: Peers per continent in week 2 of the 2016 measurement.

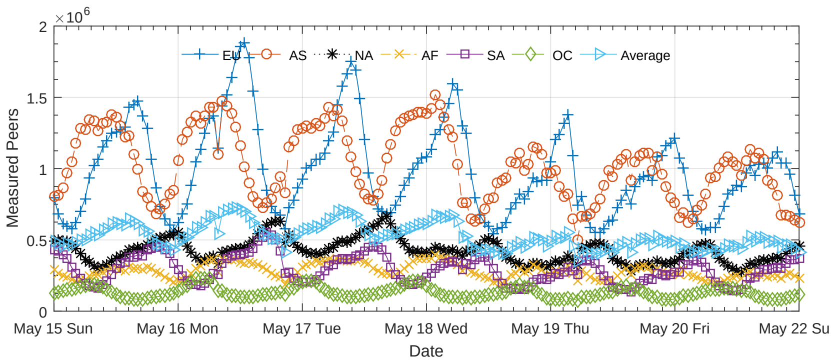


Figure B.3: Peers per continent in week 3 of the 2016 measurement.

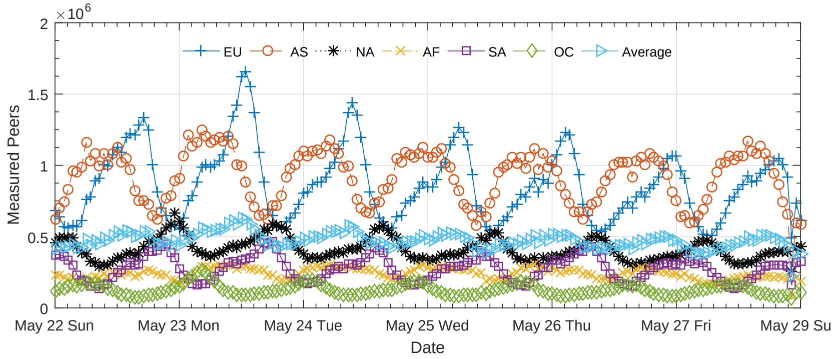


Figure B.4: Peers per continent in week 4 of the 2016 measurement.

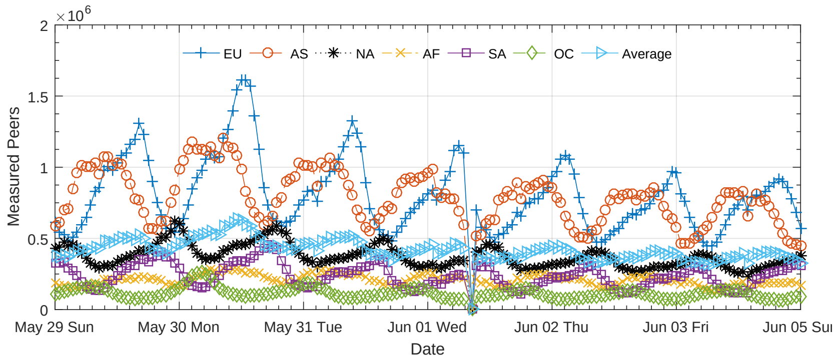


Figure B.5: Peers per continent in week 5 of the 2016 measurement.

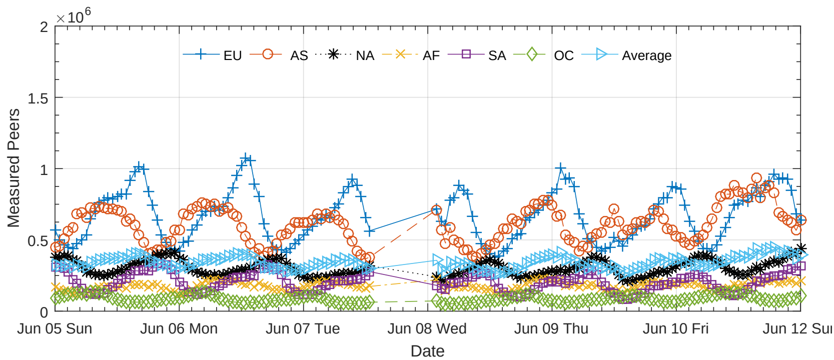


Figure B.6: Peers per continent in week 6 of the 2016 measurement.

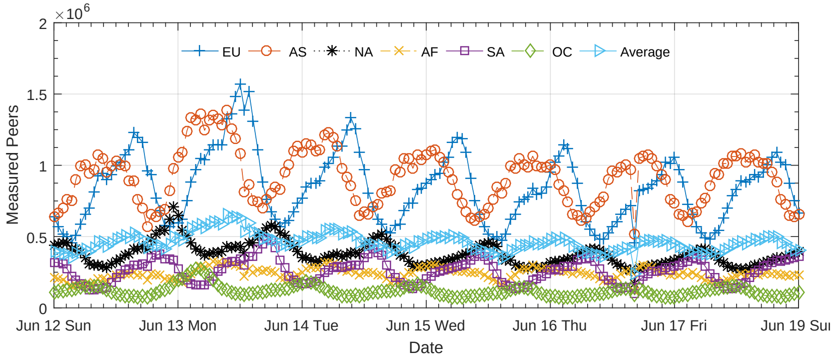


Figure B.7: Peers per continent in week 7 of the 2016 measurement.

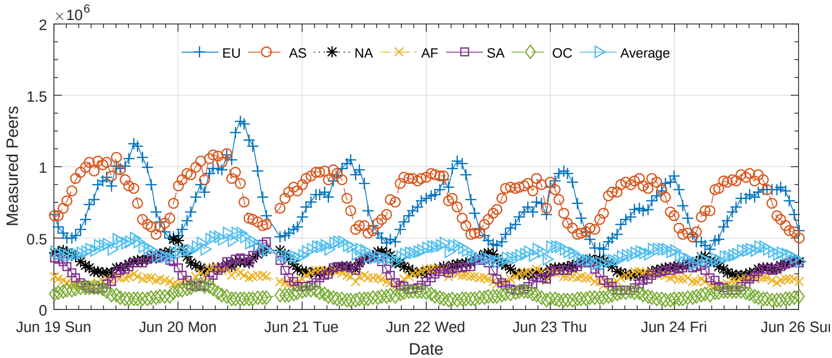


Figure B.8: Peers per continent in week 86 of the 2016 measurement.

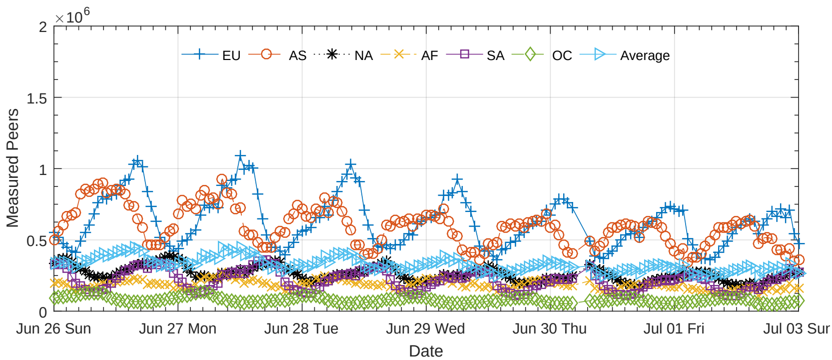


Figure B.9: Peers per continent in week 9 of the 2016 measurement.

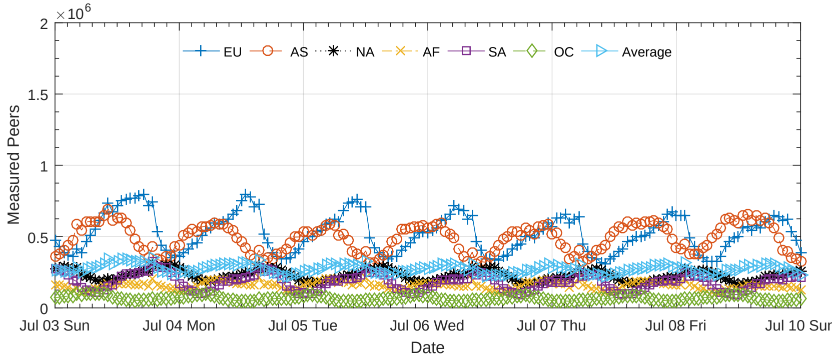


Figure B.10: Peers per continent in week 10 of the 2016 measurement.

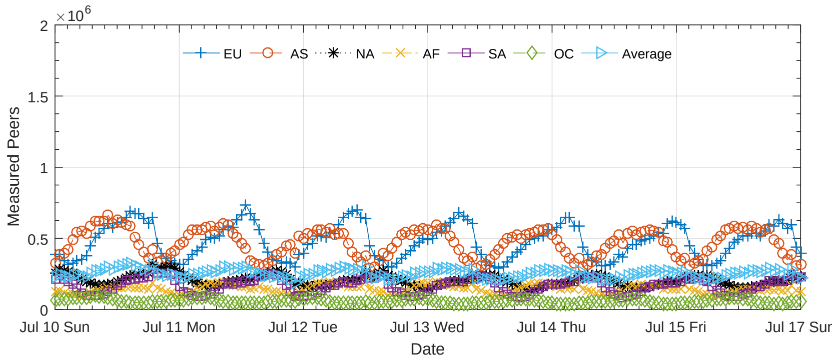


Figure B.11: Peers per continent in week 11 of the 2016 measurement.

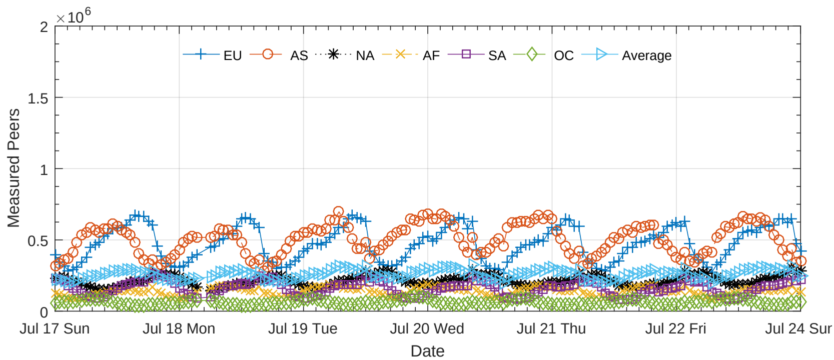


Figure B.12: Peers per continent in week 12 of the 2016 measurement.

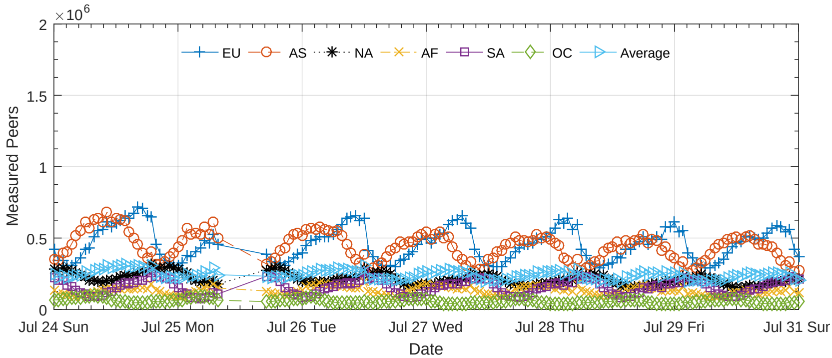
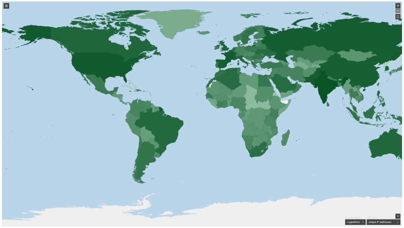


Figure B.13: Peers per continent in week 13 of the 2016 measurement.

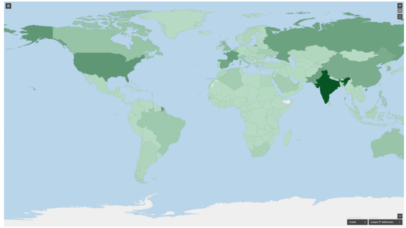
C
GeoChart

GeoChart.js (`https://github.com/GeoChart`) visualizes country-referenced numerical data on interactive choropleth maps (*i.e.* Figure C.1). GeoChart.js takes arbitrary data and produces a world map, where each country will be colored relatively to the value assigned to it. The main strength of GeoChart is the interactive interface that allows a user to select the data to be displayed and also lets a user change the color function used to determine the color of countries. The first option to use GeoChart is to feed the data directly into the view which will render the map with it. The second option is to use the provided back-end to read the data from a database following the relational data model. The GeoChart visualization system consists of three parts: a database, a back-end, and a view. The data base follows a generic data model C.2, which is queried and served to the view by the back-end. To re-use GeoChart the easiest way is to store the relevant data in a database schema following the generic data model and connect the back-end to this database. Furthermore, it is also possible to write extensions to the back-end to support other database systems and layouts or to feed the data directly to the view.

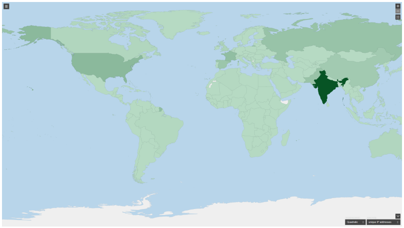
Choropleth maps visualize one value per country. The color is determined by calculating the relative position of the value in the range of *[0, maximum value]*. A specific range of two predefined colors will represent the boundaries of this range.



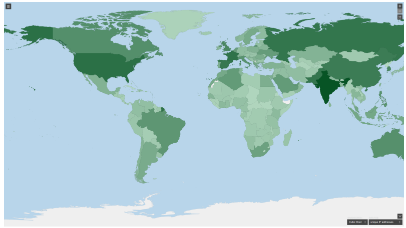
(a) Logarithmic



(b) Linear



(c) Quadratic



(d) Cubic root

Figure C.1: The same distribution of all unique IP addresses on April 13, 2015 with different color functions.

The data point will be converted into the matching color, using functions instead of classifications for the representation of values. Color functions do not suffer from some of the problems of fixed color classes, like loss of precision [1]. It is not required to define fixed ranges, but rather one function, which defines the complete course of the data range. To avoid a biased representation of the map, multiple color functions are provided. The color functions influence the relative position in the above-mentioned range of values. They were selected from the basic set of mathematical functions and can be extended easily. The logarithmic scale was chosen as the standard color function for the map since it does not react as strongly to outliers as linear functions, leading to a general improvement of the color distribution. The user is able to select the desired color function in the interactive user interface.

GeoChart supports multiple dimensions of data. It allows to represent a list of different values on the world map. Individual datasets can be easily exchanged without reloading the map. It can, for example, visualize the distribution of response times of peers as well as the sum of peers in the respective country in one instance. The current dimension's details are listed in a sidebar, which overlays the map. Each

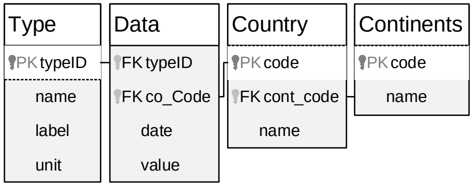


Figure C.2: GeoChart's data model. Country and Continents are static data.

country is listed with the relative value, the absolute value, and its color. The data can be downloaded into a CSV (Comma Separated Value) format to further process it. Dimensions can be switched within the sidebar overlay.

The most convenient interface offered by GeoChart is the database, which allows to provide the data to be visualized from any source with a database adaptor. Since GeoChart's data model is generic, the only requirement for using GeoChart is to transfer custom data to this generic data model C.2. The data model allows the definition of data types, which are shown in the same output map, *e.g.*, different torrents being shared at the same day.

Figure C.2 shows the data base layout. The *Type* table is used to define the types of data in the *Data* table. The *label* and *unit* attributes are displayed in the UI. The *Data* table contains the actual values per type, day, and country. The *Country* and *Continents* tables contain country and continent names that are used in the map, since this data does not change often it is delivered with GeoChart.

References

[1] G. Andrienko, N. Andrienko, A. Savinov, "Choropleth Maps: Classification Revisited," The IEEE International Conference on Communications (2001), Helsinki, Finnland, June 2001.

[2] Andy, "Netflix Chief: Piracy Prepared Internet Users For Us," https://torrentfreak.com/netflix-chief-piracy-prepared-internet-users-for-us-150605/, June 2015, [Online, accessed 2017-04-03].

[3] Andy, "Man Who Leaked The Revenant Online Fined $1.1m," https://torrentfreak.com/man-leaked-revenant-online-fined-1-1m-160930/, September 2016, [Online, accessed 2017-04-03].

[4] R. Bindal, P. Cao, W. Chan, J. Medved, G. Suwala, T. Bates, A. Zhang, "Improving Traffic Locality in BitTorrent via Biased Neighbor Selection," 26th IEEE International Conference on Distributed Computing Systems (ICDCS 2006), Lisboa, Portugal, July 2006, pp. 66–66.

[5] BitTorrent Inc., "BitTorrent Sync," https://www.getsync.com/intl/de/, [Online, accessed 2017-04-03].

[6] A. Blake, "Kickass Torrents targeted by DOJ, feds say website distributed $1 billion in copyrighted works," http://www.washingtontimes.com/news/2016/jul/20/kickass-torrents-targeted-doj-feds-say-website-dis/, July 2016, [Online, accessed 2017-04-03].

[7] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC 2007), San Diego, California, USA, October 2007.

[8] D. R. Choffnes, F. E. Bustamante, "Taming the Torrent: A Practical Approach to Reducing Cross-isp Traffic in Peer-to-peer Systems," ACM SIGCOMM 2008, Seattle, WA, USA, August 2008, pp. 363–374.

[9] Chris Morris, "Blu-Ray Struggles in the Streaming Age," http://fortune.com/2016/01/08/blu-ray-struggles-in-the-streaming-age/, January 2016, [Online, accessed 2017-04-03].

[10] CISCO, "White Paper: Cisco VNI Forecast and Methodology, 2015-2020," http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html, 2016, [Online, accessed 2017-04-03].

[11] Clark, D. and Bauer, S. and claffy, k. and Dhamdhere, A. and Huffaker, B. and Lehr, W. and Luckie, M., "Measurement and Analysis of Internet Interconnection and Congestion," Research Conference on Communication, Information and Internet Policy (TPRC 2014), Arlington, VA, USA, September 2014.

[12] B. Cohen, "The BitTorrent Protocol Specification," http://bittorrent.org/beps/bep_0003.html, [Online, accessed 2017-04-03].

[13] B. Cohen, "Incentives Build Robustness in BitTorrent," Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, USA, June 2003.

[14] S. A. Crosby, D. S. Wallach, "An Analysis of BitTorrent's Two Kademlia-Based DHTs," Department of Computer Science, Rice University, Houston, Tech. Rep. TR07-04, 2007.

[15] G. Dán, N. Carlsson, "Power-Law Revisited: Large Scale Measurement Study of P2P Content Popularity." 9th International Workshop on Perr-to-Peer systems (IPTPS 2010), San Jose, CA, USA, April 2010.

[16] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston, D. Sampath, "The YouTube Video Recommendation System," Thr Fourth ACM Conference on Recommender Systems (RecSys 2010), ser. RecSys '10, Barcelona, Spain, September 2010.

[17] C. Decker, R. Eidenbenz, R. Wattenhofer, "Exploring and Improving BitTorrent Topologies," IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P 2013), Trento, Italy, September 2013.

[18] Diana Lodderhose, "Movie piracy: threat to the future of films intensifies," https://www.theguardian.com/film/2014/jul/17/digital-piracy-film-online-counterfeit-dvds, July 2014, [Online, accessed 2017-04-03].

[19] eBizMBA Inc., "Top 15 Most Popular Torrent Sites | February 2016," http://www.ebizmba.com/articles/torrent-websites, [Online, accessed 2017-04-03].

[20] D. Elms, S. Krishna, "UK Availability of Film and TV Titles in the Digital Age," https://assets.kpmg.com/content/dam/kpmg/pdf/2015/03/uk-availability-of-film-and-tv.pdf, December 2014, [Online, accessed 2017-04-03].

[21] Ernesto, "KAT Takedown Triggers Traffic Spike at Torrent Sites," https://torrentfreak.com/kat-takedown-triggers-traffic-spike-at-torrent-sites-160724/, July 2016, [Online, accessed 2017-04-03].

[22] S. Gustin, "Here's Why Your Netflix Is Slowing Down," http://time.com/8681/netflix-verizon-peering/, February 2014, [Online, accessed 2017-04-03].

[23] T. Hoßfeld, D. Hock, S. Oechsner, F. Lehrieder, Z. Despotovic, W. Kellerer, M. Michel, "Measurement of BitTorrent Swarms and their AS Topologies," University of Würzburg, Tech. Rep. 464, Nov 2009.

[24] T. Hoßfeld, F. Lehrieder, D. Hock, S. Oechsner, Z. Despotovic, W. Kellerer, M. Michel, "Characterization of bittorrent swarms and their distribution in the internet," *Computer Networks*, Vol. 55, No. 5, pp. 1197–1215, 2011.

[25] ICT Data and Statistics Division, "ICT Facts & Figures," https://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2015.pdf, May 2015, [Online, accessed 2017-04-03].

[26] IMDb.com, Inc., "KAT Takedown Triggers Traffic Spike at Torrent Sites," http://www.imdb.com/title/tt4077554/?ref_=ttep_ep2, May 2016, [Online, accessed 2017-04-03].

[27] N. Jurran, "Hinter den Kulissen von Netflix," http://heise.de/-2304264, August 2014, [Online, accessed 2017-04-03].

[28] S. Kaune, R. C. Rumin, G. Tyson, A. Mauthe, C. Guerrero, R. Steinmetz, "Unraveling BitTorrent's File Unavailability: Measurements and Analysis," IEEE Tenth International Conference on Peer-to-Peer Computing (P2P 2010), Delft, The Netherlands, August 2010.

[29] Kickass Torrents, "Deadpool 2016 1080p BluRay x264 DTS-JYK," https://kat.cr/deadpool-2016-1080p-bluray-x264-dts-jyk-t12504043.html#main, [Online, accessed 2017-04-03].

[30] KickassTorrents, "KickassTorrents - Kickass - Download Torrents from Kickass Torrents," http://kickasstorrents.to/, [Online, accessed 2017-04-03].

[31] M. Kryczka, R. Cuevas, C. Guerrero, A. Azcorra, A. Cuevas, "Measuring the bittorrent ecosystem: Techniques, tips, and tricks," *Communications Magazine, IEEE*, Vol. 49, No. 9, pp. 144–152, 2011.

[32] E. Langford, R. Langford, "Solution of the Inverse Coupon Collector's Problem," *The Mathematical Scientist*, Vol. 27, No. 2, December 2002, [Online, accessed 2016-11-25].

[33] A. Lareida, T. Bocek, M. Pernebayev, B. Stiller, "Automatic Network Configuration with Dynamic Churn Prediction," IFIP/IEEE International Symposium on Integrated Network Management (IM 2015), Ottawa, ON, Canada, May 2015.

[34] A. Lareida, T. Bocek, M. Waldburger, B. Stiller, "RB-Tracker: A Fully Distributed, Replicating, Network-, and Topology-aware P2P CDN," 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, Belgium, May 2013.

[35] A. Lareida, G. Petropoulos, V. Burger, M. Seufert, S. Soursos, B. Stiller, "Augmenting home routers for socially-aware traffic management," Local Computer Networks (LCN), 2015 IEEE 40th Conference on, Clearwater Beach, FL, USA, Oct 2015.

[36] A. Lareida, T. Hoßfeld, B. Stiller, "The BitTorrent Peer Collector Data Set," http://www.csg.uzh.ch/publications/data/peercollector.html, [Online, accessed 2017-04-03].

[37] A. Lareida, T. Hoßfled, B. Stiller, "The BitTorrent Peer Collector Problem," IEEE/IFIP Network Operations and Management Symposium, Lisbon, Portugal, May 2017, pp. 1–7.

[38] A. Lareida, R. Pernischova, B. B. Rodrigues, B. Stiller, "Abstracting .torrent Content Consumption into Two-mode Graphs and their Projection to Content Networks (ConNet)," 2017 IFIP/IEEE International Symposium on Integrated Network Management (IM). Lisbon, Portugal: IEEE, May 2017, pp. 1–9.

[39] A. Lareida, S. Schrepfer, T. Bocek, B. Stiller, "Overlay network measurements with distribution evolution and geographical visualization," IEEE/IFIP Network Operations and Management Symposium. Istanbul, Turkey: IEEE, Apr 2016, pp. 1–9.

[40] A. Loewenstern, A. Norberg, "Dht protocol," http://bittorrent.org/beps/bep_0005.html, [Online, accessed 2017-04-03].

[41] G. Maier, A. Feldmann, V. Paxson, M. Allman, "On Dominant Characteristics of Residential Broadband Internet Traffic," 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC 2009), Chicago, Illinois, USA, Nov 2009.

[42] R. B. Mansilha, L. R. Bays, M. B. Lehmann, A. Mezzomo, G. Facchini, L. P. Gaspary, M. P. Barcellos, "Observing the BitTorrent Universe Through Telescopes," 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops, Dublin, Ireland, May 2011.

[43] P. Maymounkov, D. Mazieres, "Kademlia: A peer-to-peer information system based on the XOR metric," 1st International Workshop on Peer-to-Peer Systems (IPTPS 2002), Cambridge, MA, USA, March 2002.

[44] P. Maymounkov, D. Mazières, *Peer-to-Peer Systems: First International Workshop, IPTPS 2002 Cambridge, MA, USA, March 7–8, 2002 Revised Papers*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, ch. Kademlia: A Peer-to-Peer Information System Based on the XOR Metric, pp. 53–65. [Online]. Available: http://dx.doi.org/10.1007/3-540-45748-8_5

[45] Netflix Inc., "Netflix Open Connect," https://openconnect.netflix.com/en_gb/, 2016, [Online, accessed 2017-04-03].

[46] W. B. Norton, "Internet Service Providers and Peering," 21st North American Network Operators' Group Meeting (NANOG, Oakland, CA, USA, October 2001.

[47] S. Oechsner, F. Lehrieder, T. Hoßfeld, F. Metzger, D. Staehle, K. Pussep, "Pushing the Performance of Biased Neighbor Selection through Biased Unchoking," IEEE Ninth International Conference on Peer-to-Peer Computing (P2P 09), Seattle, WA, USA, September 2009, pp. 301–310.

[48] OpenTrackr, "Free to use Torrent tracker!" http://opentrackr.org/, [Online, accessed 2017-04-03].

[49] Oracle, "Package java.nio," https://docs.oracle.com/javase/7/docs/api/java/nio/package-summary.html, [Online, accessed 2017-04-03].

[50] Oracle Corporation and/or its affiliates, "MySQL," https://www.mysql.com, [Online, accessed 2017-04-03].

[51] J. S. Otto, M. A. Sánchez, D. R. Choffnes, F. E. Bustamante, G. Siganos, "On Blind Mice and the Elephant: Understanding the Network Impact of a Large Distributed System," ACM SIGCOMM 2011, Toronto, Ontario, Canada, Aug 2011.

[52] N. Rego, "Exclusive: Netflix's CEO Talks Competition, VPNs, and the Future of Streaming," http://www.tbreak.com/exclusive-netflixs-ceo-talks-competition-vpns-and-the-future-of-streaming, April 2016, [Online, accessed 2017-04-03].

[53] R. C. Rumin, N. Laoutaris, X. Yang, G. Siganos, P. Rodriguez, "Deep Diving into BitTorrent Locality," IEEE INFOCOM 2011, Shanghai, China, April 2011.

[54] Sandvine Incorporated ULC, "2015 Global Internet Phenomena - Asia-Pacific & Europe," https://www.sandvine.com/downloads/general/global-internet-phenomena/2015/global-internet-phenomena-report-apac-and-europe.pdf, 2015, [Online, accessed 2017-04-03].

[55] Sandvine Incorporated ULC, "2016 Global Internet Phenomena - Latin America & North America," https://www.sandvine.com/downloads/general/global-internet-phenomena/2016/global-internet-phenomena-report-latin-america-and-north-america.pdf, 2016, [Online, accessed 2017-04-03].

[56] S. Saroiu, P. K. Gummadi, S. D. Gribble, "A measurement study of peer-to-peer file sharing systems," *SIGCOMM Computer Communication Review*, Vol. 32, No. 1, pp. 82–82, Dec 2002.

[57] J. Scott, *Social network analysis*. Sage, 2012.

[58] H. Steier, "Warum Swisscom-Kunden Probleme mit Netflix hatten," http://www.nzz.ch/digital/streaming-warum-swisscom-kunden-probleme-mit-netflix-haben-ld.9174, March 2016, [Online, accessed 2017-04-03].

[59] R. Steinmetz, K. Wehrle, "2. what is this "peer-to-peer" about?" in *Peer-to-Peer Systems and Applications*, ser. Lecture Notes in Computer Science, R. Steinmetz, K. Wehrle, Eds. Springer Berlin Heidelberg, 2005, Vol. 3485, pp. 9–16. [Online]. Available: http://dx.doi.org/10.1007/11530657_2

[60] The 8472, "Peer Exchange (PEX)," http://www.bittorrent.org/beps/bep_0011.html, [Online, accessed 2017-04-03].

[61] The Apache Software Foundation, "Apache Avro 1.8.1 Documentation," https://avro.apache.org/docs/current/, [Online, accessed 2017-04-03].

[62] The Apache Software Foundation, "Apache Hive," https://hive.apache.org/, [Online, accessed 2017-04-03].

[63] The Apache Software Foundation, "Apache Parquet," https://parquet.apache.org/, [Online, accessed 2017-04-03].

[64] The Apache Software Foundation, "Apache Spark - Lightning-fast cluster computing ," http://spark.apache.org/, [Online, accessed 2017-04-03].

[65] The Apache Software Foundation, "Welcome to Apache Hadoop!" http://hadoop.apache.org/, [Online, accessed 2017-04-03].

[66] The igraph Core Team, "igraph - The Network Analysis Package," http://igraph.org/, [Online, accessed 2017-04-03].

[67] The Pirate Bay, "Download Music, Movies, Games, Software! The Pirate Bay - The Galaxy's Most Resilient BitTorrent Site," https://thepiratebay.se/, [Online, accessed 2017-04-03].

[68] theory.org, "Bittorrent peer exchange conventions," https://wiki.theory.org/index.php?title=BitTorrentPeerExchangeConventions&oldid=2707, [Online, accessed 2017-04-03].

[69] theory.org, "Bittorrent Protocol Specification v1.0," https://wiki.theory.org/index.php?title=BitTorrentSpecification&oldid=2962, [Online, accessed 2017-04-03].

[70] Valdis Krebs, "Social Network Analysis: An Introduction," http://www.orgnet.com/sna.html, [Online, accessed 2017-04-03].

[71] F. Vogt, "Exklusiv: Netflix vs. Swisscom: Swisscom Knickt ein, Peering-Vertrag Unterzeichnet," http://www.computerworld.ch/news/it-branche/artikel/exklusiv-netflix-vs-swisscom-swisscom-knickt-ein-peering-vertrag-unterzeichnet-69890/, March 2016, [Online, accessed 2017-04-03].

[72] Vuze Inc., "Distributed Hash Table," http://wiki.vuze.com/mediawiki/index.php?title=Distributed_hash_table&oldid=14796, [Online, accessed 2017-04-03].

[73] K. Wajda, R. Stankiewicz, Z. Duliński, T. Hoßfeld, M. Seufert, D. Hausheer, M. Wichtlhuber, I. Papafili, M. Dramitinos, P. Cruschelli *et al.*, "Socially-Aware Management of New Overlay Applications Traffic - The Optimization Potentials of the SmartenIT Approach," International Conference on Mobile Networks and Management, Würzburg, Germany, September 2014.

[74] D. Wu, P. Dhungel, X. Hei, C. Zhang, K. Ross, "Understanding Peer Exchange in BitTorrent Systems," IEEE Tenth International Conference on Peer-to-Peer Computing (P2P 2010), Delft, The Netherlands, August 2010.

[75] W. Xu, A. K. Tang, "A Generalized Coupon Collector Problem," *Journal of Applied Probability*, Vol. 48, No. 4, December 2011.

[76] C. Zhang, P. Dhungel, D. Wu, K. W. Ross, "Unraveling the BitTorrent Ecosystem," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 22, No. 7, pp. 1164–1177, 2011.

Other Author Publications

A. Lareida, D. Meier, T. Bocek, and B. Stiller, “Towards path quality metrics for overlay networks,” 41st Annual IEEE Conference on Local Computer Networks (LCN). Dubai, United Arab Emirates, November 2016.

G. Petropoulos, A. Lareida, V. Burger, M. Seufert, S. Soursos, and B. Stiller, “Wifi offloading and socially aware prefetching on augmented home routers,” 40th Conference on Local Computer Networks (LCN) 2015. Clearwater Beach, FL, USA, October 2015.

A. Lareida, T. Bocek, S. Golaszewski, C. Lüthold, and M. Weber, “Box2box - a p2p-based file-sharing and synchronization application,” Peer-to-Peer. Trento, Italy, September 2013.

Curriculum Vitae

Andri Filip Lareida was born in Aarau July 14, 1984. Before starting his academic career, he visited primary school in Klosters Switzerland. After a move to Bad Zurzach, where Andri completed secondary school, he went to the Kantonsschule Wettingen to receive his Matura.

Academic Education

7.12–6.17 PhD in Informatics at University of Zurich Faculty of Economics, Business Administration and Information Technology Switzerland. Specialized in distributed systems, measurements, data analysis and processing

2.10–4.12 MSc in Informatics at University of Zurich Faculty of Economics, Business Administration and Information Technology Switzerland. Specialized in software systems. Title of Thesis: "Development and Implementation of the B-Tracker Approach on a BitTorrent Client".

2.10–6.10 ERASMUS-Semester University of Aberdeen The School of Natural and Computing Sciences Scotland

9.00–2.07 BSc in Informatics University of Zurich Faculty of Economics, Business Administration and Information Technology Switzerland. Specialized in information systems. Title of thesis: "Development of a GUI for Self-Assembly Simulator Configuration"

Employment History

7.12–5.17 Junior Researcher. 80% University of Zurich, Institute for Informatics, Switzerland

6.08–10.11 Workstudent 60%, IBM Schweiz AG, Zurich, Switzerland

10.06–5.08 Workstudent 40% IBM Schweiz AG, Zurich, Switzerland

6.08–10.11 Intern IBM Schweiz AG, Zurich, Switzerland